Visual Servoing For Robotic Positioning And Tracking Systems

Yimin Zhao

A Thesis

In the Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of

Doctor of Philosophy (Mechanical and Industrial Engineering) at

Concordia University

Montreal, Quebec, Canada

August 2012

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By **Yimin Zhao**

Entitled **Visual Servoing for Robotic Positioning and Tracking Systems**

and submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY (Mechanical Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____Chair
Dr. D. Dysart-Gale

_____External to Program
Dr. C.Q. Wu

_____ Examiner
Dr. A. Ben Hamza

_____ Examiner
Dr. H. Hong

_____ Examiner
Dr. J. Dargahi

_____Thesis Supervisor
Dr. W. Xie

Approved by

_____Chair of Department or Graduate Program Director
Dr. A. Dolatabadi

_____Dean of Faculty of Engineering and Computer Science
Dr. Robin A.L. Drew

November 30, 2012

ABSTRACT

Visual Servoing for Robotic Positioning and Tracking Systems

Yimin Zhao, Ph.D.
Concordia University, 2012

Visual servoing is a robot control method in which camera sensors are used inside the control loop and visual feedback is introduced into the robot control loop to enhance the robot control performance in accomplishing tasks in unstructured environments. In general, visual servoing can be categorized into image-based visual servoing (IBVS), position-based visual servoing (PBVS), and hybrid approach. To improve the performance and robustness of visual servoing systems, the research on IBVS for robotic positioning and tracking systems mainly focuses on aspects of camera configuration, image features, pose estimation, and depth determination.

In the first part of this research, two novel multiple camera configurations of visual servoing systems are proposed for robotic manufacturing systems for positioning large-scale workpieces. The main advantage of these two multiple camera configurations is that the depths of target objects or target features are constant or can be determined precisely by using computer vision. Hence the accuracy of the interaction matrix is guaranteed, and thus the positioning performances of visual servoing systems can be improved remarkably. The simulation results show that the proposed multiple camera configurations of visual servoing for large-scale manufacturing systems can satisfy the demand of high-precision positioning and assembly in the aerospace industry.

In the second part of this research, two improved image features for planar central symmetrical-shaped objects are proposed based on image moment invariants, which can represent the pose of target objects with respect to camera frame. A visual servoing controller based on the proposed image moment features is designed and thus the control performance of the robotic tracking system is improved compared with the method based on the commonly used image moment features. Experimental results on a 6-DOF robot visual servoing system demonstrate the efficiency of the proposed method.

Lastly, to address the challenge of choosing proper image features for planar objects to get maximal decoupled structure of the interaction matrix, the neural network (NN) is applied as the estimator of target object poses with respect to camera frame based on the image moment invariants. Compared with previous methods, this scheme avoids image interaction matrix singularity and image local minima in IBVS. Furthermore, the analytical form of depth computation is given by using classical geometrical primitives and image moment invariants. A visual servoing controller is designed and the tracking performance is enhanced for robotic tracking systems. Experimental results on a 6-DOF robot system are provided to illustrate the effectiveness of the proposed scheme.

ACKNOWLEDGEMENTS

I am sincerely grateful to my supervisor, **Dr. Wen-Fang Xie** for her initiation of the research project, her guidance and assistance, as well as her invaluable support throughout this research. Her supervision and encouragement were essential to my completing this work. Her patience and persistence made all achievements on this research project possible.

I am extremely grateful to the Aerospace Manufacturing Technology Centre (AMTC), and the National Research Council of Canada (NRC), which have provided facilities for this research and a great opportunity to access the visual servoing research filed. Especially, I would like to thank Dr. Xiao-Wei Tu and the robotics automation group of AMTC for their continuous support, guidance and assistance.

I thank my colleagues and friends at Concordia University who have offered help in various ways. I am especially grateful to Mr. Abolfazl Mohebbi, his contribution to the robot hardware greatly assisted in the experiments of this research. I am very grateful to Mr. Gilles Huard, his kind help and perfect work on the experiment setup of the research enabled smooth and successful results.

I would like to express my gratitude to my parents, my brother and sister for their enormous encouragement and support. Last, but not least, I would like to express special thanks to my dear wife, Fangxia Lin. It is her love, help, understanding and patience that made it possible for me to accomplish this research throughout my Ph.D. study. Thanks to my two sons, Yue Zhao and Willams Daniel Zhao, for their encouragement, support and love. I dedicate this thesis to my wife and my family.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

SYMBOL	DESCRIPTION
$\{B\}$	Robot base frame
$\{E\}$	End-effector frame
$\{C_i\}$	The frame of camera i $(i = 1,2,3,4)$
${}^{B}_{E}T$	The transformation matrix from $\{E\}$ to $\{B\}$
${}^{E}_{C_i}T$	The transformation matrix from $\{C_i\}$ $(i = 1,2,3,4)$ to $\{E\}$
ξ	Parametric variable
$\dot{r}_{C_i}$	Velocity screw of the camera i expressed in $\{C_i\}$
$\dot{r}$	Velocity screw of robot end-effector expressed in $\{E\}$
${}^{E}_{C_i}T_v$	Velocity transformation matrix from $\{C_i\}$ $(i = 1,2,3,4)$ to $\{E\}$
f_i	Image features in the image plane of camera i $(i = 1,2,3,4)$
$\dot{f}_i$	Velocity of image features in the image plane of camera i $(i = 1,2,3,4)$
λ	Focal length
(α_x, α_y)	The effective size of a pixel of camera image plane
K	Proportional gain
q	Vector of joint variable
$\dot{q}$	Vector of joint velocity
$\ddot{q}$	Velocity of joint acceleration

$\dot{q}_d$	Desired robot joint velocity determined by visual servoing law
w_e	Wrench applied by environment to robot end-effector
τ	Control vector of joint torques sent to robot
τ_e	Vector of joint torques corresponding to an external wrench w_e
$\widetilde{\tau}_e$	The computed vector of torques by robot toolbox
${}^{E}_{C}T_f$	Force-moment transformation matrix from $\{C\}$ to $\{E\}$
K_p	Proportional coefficient matrix
K_v	Differential coefficient matrix
$J(q)$	Robot velocity Jacobian matrix
J_{image}	Interaction matrix or image Jocobian matrix
β	Rotational angle around *x* axis of camera frame
γ	Rotational angle around *x* axis of camera frame
$\hat{\beta}$	Estimation values of rotational angle around *x* axis of camera frame
$\hat{\lambda}$	Estimation values of rotational angle around *x* axis of camera frame
ϕ	The orientation angle
m_{ij}	Image moment of order $i+j$
μ_{ij}	Image central moment of order $i+j$
(x_g, y_g)	Coordinates of centroid of image

u	Feedback control signal

CHAPTER 1 INTRODUCTION

1.1.1 Robotic positioning and tracking systems

In the aerospace industry, the tasks in fastening large-scale parts such as drilling, riveting and welding, in assembly such as piping and aircraft wing alignment need high precision positioning. It is difficult and expensive to obtain a robot with high payload using traditional methods. Although 3D metrology and tracking systems including laser tracker, laser radar [1], indoor GPS (Global Positioning System) [2], and other optic-electronic positioning systems are used for positioning in manufacturing and assembly, these systems are highly limited in terms of speed and workspace, and generally very expensive. One feasible and cost-effective way to overcome these constrains is to use a visual servoing method to achieve high precision positioning.

Visual servoing is a robot control method in which visual feedback or image from camera sensors is introduced into the robot inside a control loop to accomplish tasks in unstructured environments and to enhance the robot control performance [3, 4]. Visual servoing can be categorized into image-based visual servoing (IBVS), position-based visual servoing (PBVS), and hybrid approach [5]. In the visual servoing systems, there are two basic camera configurations. One is to install a camera at the tip or end-effector of the manipulator (eye-in-hand); the other is to set the camera and manipulator separately (eye-to-hand) [6].

Compared with the other visual servoing methods, IBVS has three main advantages [7, 8, 9] and has gained research interest among the robotics community since it is insensitive and robust to camera model error, calibration error, and measurement noise [10]. Hence IBVS is widely used for automated robotic manufacturing systems [11].

However, it is difficult for traditional IBVS with a single camera to deal with the tasks of robot positioning for large-scale manufacturing systems in the aerospace industry [12]. The reason is that the target features of a large size workpiece are chosen far away from each other to increase the accuracy of positioning in large-scale manufacturing systems. It is obvious that it is difficult, if not impossible, to place all target features in the field of view (FOV) of a single camera. Moreover, due to the large size of the workpiece, the disposal of cameras is a key problem [13]. Sometimes the target features are occluded from the FOV of the camera, wherein an unsuitable viewpoint may cause some target features to move out of the FOV of the camera during visual servoing. This may lead to servoing failure, especially when the initial position of the camera is far away from its desired one [14, 15]. Hence, some researchers have proposed new configurations for cameras to improve the visibility of target objects, such as using stereo cameras [16], or multiple cameras [17, 18]. Nevertheless, these methods are only effective in a situation where the size of the target object is relatively small and the robot end-effector moves in a large range. However, in the aerospace industry, a robot end-effector generally moves in the vicinity of the desired position and the target features are chosen far away from each other in visual servoing. Therefore, the conventional configurations of cameras in visual servoing cannot effectively deal with the task of robot positioning for large-scale manufacturing systems.

In addition, it is known that the interaction matrices of IBVS actually depend on the depths of target features [19]. In most of cases, target feature depth is treated as constant or just roughly estimated such that the visual servoing system remains stable and convergent because of the robustness of IBVS to the error of camera model and system

calibration. Nevertheless, in some cases the incorrect depth estimation of target features may cause convergence and stability problems [20]. Therefore, the exact determination of target feature depths is still a crucial task in the design of IBVS systems.

Moreover, in large-scale manufacturing systems, the dynamic forces of the workpiece play an important role in robot dynamic behaviors, since the mass of large size workpiece is not negligible. However, most papers published on visual servoing focus only on the kinematic analysis of the visual servoing systems. The dynamics of the robot is hardly taken into consideration in the visual servoing scheme design.

In this thesis, multiple camera visual servoing for large-scale manufacturing systems is investigated, and the dynamics of robot manufacturing systems are taken into consideration in designing IBVS system controller. Two novel configurations of multiple camera visual servoing for a large-scale 3D positioning system are proposed, which will be discussed in Chapter 3 and Chapter 4. The list of titles is as follows:

1. "Multiple camera visual servoing with 3D CAD laser projector for large-scale 3D positioning"
2. "Multiple camera-multiple target point visual servoing system for large scale 3D positioning"

In visual servoing, one of the drawbacks of IBVS is that there exist interaction matrix singularities and image local minima leading to IBVS failure in robot workspace [21]. Hence the choice of target features is another key point to address the challenge of interaction matrix singularities. Much effort have been contributed to determine some decoupling image features to deliver a triangular or diagonal interaction matrix [22, 23].

In the determination of image features in visual servoing, geometric features such as

points, segments or straight lines [3] are usually chosen as the target features, and the corresponding image features in the image plane are utilized as image features and the inputs of visual servoing controllers. However, such target features can only be applied in certain limited target objects [24]. In addition, the geometric features might be occluded from the FOV of the camera. In this case, the number of the image features does not match with that of the desired ones, which may lead to the failure of visual servoing. In order to track the target objects which do not have enough detectable geometric features, and to enhance the robustness of visual servoing systems, several novel target features are adopted for visual servoing. For example, laser points [13, 25] and the polar signature of target object contour [26] are used as target features in the design of IBVS systems.

The image moments are normally used for pattern-recognition in computer vision [27, 28]. Recently they have been adopted as image features for viusal servoing control scheme design due to their easy computation from binary or segmented image or from a set of extracted points of interest, disregarding the target object shape complexity and their generic representation of any target object, with a simple or complex shape [29]. In addition, low-order moments have an intuitive meaning, since they are directly related to the area, the centroid, and the orientation of the object in the image plane [21]. Following previous work, it is known that using the image moments as image features in visual servoing can render the corresponding interaction matrix with maximal decoupled structure, and the inherent problem-singularity of the interaction matrix is avoided and the control performance of IBVS system is thus improved.

In [22], based on image moment invariants to 2D translation, 2D rotation, and scale, two image features are selected to control ω_x, ω_y the rotational velocity around the x and

y axes of the camera frame respectively for planar non-centered symmetrical objects. But these two image features cannot be used for planar central symmetrical objects since the elements of the interaction matrix related to these two image features are equal to zero when the planar target object is parallel to the image plane [21]. In this case, two other image features S_x, S_y are proposed to control the rotational velocities ω_x, ω_y for planar central symmetrical objects in order to avoid the singularity of the interaction matrix. However, the simulation results show that image features S_x, S_y for planar central symmetrical objects used in [22] cannot represent the pose of planar target objects correctly with respect with camera frame all the time. In this thesis, two improved image moments for planar central symmetrical objects are proposed to control target object pose of rotating around the *x* and *y* axes respectively. Along with the other four common image moments in [21, 22], how to analytically derive the interaction matrix describing the relationship between the motion of camera and the derivative of the image features with respect to time is illustrated. To control the motion of the camera, an IBVS controller is developed for object tracking by using the derived interaction matrix. The developed controller is applied to visual servoing for a 6-DOF robotic tracking system and the experimental results demonstrate the effectiveness of the proposed method. The details are given in Chapter 5.

Moreover, it is known that the judicious combinations of image moments are invariant to some transformations, such as 2D translation, 2D rotation, and scale. This property is of great value in visual servoing [29]. No matter what the nature of the possible image features extracted from the image is, from the coordinates of a set of image features to a set of image moments, the main question lies in how to combine them to obtain an

adequate structure of the interaction matrix of the visual servoing system. The method of the reported combinations is just stacking [21, 22]. However, the inappropriate combination of image moments and redundant image point coordinates may cause some potential problems such as local minima and coupled target features that lead to inadequate robot trajectories [28]. Some researchers have tried to use different image moments as image features to solve this problem [21, 22, 28]. In [28], the 2D image moments are combined as image features to decouple the control of the rotation from that of the translation to ensure the visibility of the target object in the FOV of the camera and also to demonstrate the global stability of the system using only the measurements from the current and desired images. In [30, 31], the choice of image features lets the condition number of interaction matrix be small to improve the robustness and the stability of the system. In [32], the image moment invariants to 2D translational motion, to 2D rotational motion and to scale were exploited to decouple translational velocity v_x, v_y, v_z from rotational velocity ω_z. In [22], several combinations of image moment invariants have been proposed and two of them are selected as image features to control rotational velocity ω_x, ω_y depending on the object considered. Such a selection of two independent image features makes the corresponding interaction matrix have a maximal decoupled structure. Indeed, when the object is parallel to image plane, the interaction matrix relating camera motion to variations of these image features, denoted as $J_{\parallel}$, has the expected form $J_{\parallel} = \begin{bmatrix} 0 & 0 & 0 & L_{\omega x} & L_{\omega y} & 0 \end{bmatrix}$. In the experimental phase of [28], the error: $e_c(\beta,\gamma) = (c_{i_t}(\beta,\gamma) - c_{i_t}^*)^2 + (c_{j_t}(\beta,\gamma) - c_{j_t}^*)^2$ was proposed to select the pair (c_i, c_j) such that the error e_c presents the global minima, with an influence zone that is as large

and symmetrical as possible. This offline selection process has to be performed for each new object considered, once the desired image is acquired. Nevertheless, it is noticed that in the interaction matrices in [22, 28], the rotational velocity ω_x , ω_y are still not decoupled. Furthermore, in the previous research, the depths of target features are assumed to be constant [21] or approximately computed from the planar object equation expressed in desired position [22, 28]. This assumption simplifies the mathematical development and such an approximation is generally accepted in practice because of the robustness of visual servoing to modeling errors. However, for large displacement visual servoing, especially in robotic tracking systems, this assumption will cause a computation error and possible singularity problems. Therefore, the exact determination of depths online is a key step in the design of visual servoing of high precision robotic tracking systems.

In this thesis, to decouple the components corresponding to ω_x , ω_y in interaction matrix, neural network (NN)-based image features are proposed, which are referred to as virtual image features and meanwhile are assumed to be proportional to the rotational angles around the *x* and *y* axes of camera frame respectively. A NN is designed to estimate the rotational angles around the *x* and *y* axes of the camera frame. Then the interaction matrix related to the proposed image features can be determined based on the outputs of the NN. In addition, the depths of target features of the planar-shaped object are determined by using image moment invariants online. Hence the computation accuracy of the interaction matrix of the visual servoing system is remarkably improved. The effectiveness of the proposed method is demonstrated in the experimental results on a 6-DOF robotic tracking system. The details are given in Chapter 6.

1.1.2 Multiple camera visual servoing for large-scale manufacturing systems

In this thesis, the multiple camera configurations of visual servoing for large-scale manufacturing are proposed. IBVS is used as visual servoing strategy and eye-in-hand architecture is adopted as camera configuration. Meanwhile, it has been proven that at least four target points are needed to have the complete velocity screw of robot end-effector uniquely defined through pseudo inverse of interaction matrix [33]. Hence, a precise robotic 3D positioning system, which is shown in Figure 3-1, is proposed by using a 3D CAD laser projector, four digital cameras and four projection screens attached to a bracket installed on a robot end-effector or workpiece, and a 6-DOF robot. In this system, a 3D CAD laser projector projects computer images (typically from CAD files) directly on objects for layout and alignment applications [34], which can provide 4 target points on the projection screens at different working positions. Four cameras are used as visual sensors, and each camera can only look at one target point on the corresponding projection screen. During the visual servoing process, each target point is always within the FOV of the corresponding camera. It is noticed that, at one working position, the laser beams remain stationary in the robot base frame, and the target points (the laser spots on the projection screens) will move along the laser beams if the robot end-effector moves in the robot base frame. Nevertheless, the depths of target points will always be kept constant during the visual servoing process because of the unique system configuration, i.e., the position of each projection screen with respect to the corresponding camera is fixed.

However, in the above multiple camera configuration of visual servoing system, the 3D CAD laser projector is very expensive. Sometimes the laser beams are occluded by objects, e.g., robot arm. In addition, the calibration of the system is also time-consuming. Hence, a new multiple camera configuration of visual servoing system shown in Figure 4-3 is proposed, in which four pairs of LEDs are installed in the robot base frame right above the cameras attached to a bracket installed on the robot end-effector or workpiece. In each pair of LEDs, one of LEDs is used as the main target point (target feature), and the other is used as the complimentary target point, which is used to determine the depth of target feature in visual servoing. Meanwhile, in visual servoing, each camera only observes one pair of LEDs. This configuration ensures that all the target points are visible when the robot end-effector moves in a certain limit range around the desired position. The depths of target points can be easily computed online by using the stereo vision strategy.

In addition, the dynamic model of the robot manufacturing system is considered in this thesis. Visual servoing control schemes, including visual servoing controller and robot dynamic controller, are designed for multiple camera configurations of visual servoing system based on machine vision, kinematics, dynamics, control theory as well as robotics. Simulations have been carried out to verify the effectiveness of the proposed visual servoing schemes under various conditions. The image errors of visual servoing systems are analysed and the stability condition of the visual servoing system is given based on the control theory. The simulation results also demonstrate that the proposed systems can achieve high-precision positioning for large-scale manufacturing and assembly systems.

1.1.3 Visual servoing by image moments

Firstly, based on the previous research on the image moment-based visual servoing, the two improved image moments, which are closely related to the pose of the target object rotating around the *x* and *y* axes of the camera frame respectively, are proposed as image features for planar central symmetrical objects. Along with the other four commonly-used image moments, the interaction matrix describing the relationship between the motion of the camera and the velocities of image features is derived, and the IBVS controller is developed for visual servoing of a 6-DOF robotic tracking systems. The simulation and experimental results demonstrate the effectiveness of the proposed methods.

Secondly, to achieve the goal of decoupling ω_x, ω_y, two virtual image features are proposed, which are assumed to be proportional to the angles of the target object rotating around the *x* and *y* axes of the camera frame respectively. Based on the image moment invariants, the NN is designed to estimate the rotational angles of a target object around the *x* and *y* axes of the camera frame. The interaction matrix related to the proposed target features is determined based on the estimation results of the NN. The depth of the planar-shaped target object can be computed online and thus the computation accuracy of the interaction matrix as well as the global stability of the system are improved. The effectiveness of the proposed method is demonstrated in the experimental results on a 6-DOF robot tracking system.

Outline of this thesis

The dissertation is composed of seven chapters. The introduction chapter introduces basic concepts and background research. The motivation, objective and contribution of this research are presented as well.

In Chapter 2, a detailed literature review is conducted for visual servoing for robotic positioning and tracking systems, followed by how the present research advances the knowledge in the literature.

Based on a 3D CAD laser projector, the multiple camera visual servoing used for large-scale 3D positioning system is proposed in Chapter 3.

In Chapter 4, multiple camera-multiple target point visual servoing in large-scale 3D manufacturing systems is presented.

To deal with the robotic tracking system, the improved image moment's visual servoing is discussed in Chapter 5.

In Chapter 6, neural network-based image moments visual servoing for robotic tracking systems is provided.

Chapter 7 contains the major conclusions of this dissertation and recommendations for future work.

CHAPTER 2 LITERATURE REVIEW

2.1.1 History of visual servoing

Robot visual servoing has been a very active research subject for the past 30 years. With the progress in electronic hardware, the robot guided by optic-electronic sensors or more recently by a machine vision system is realized. In industrial applications, the robotic systems from a simple "pick and place" robot to an advanced manufacturing robot-team have been implemented. The term "vision servoing**"** appears to have been introduced by Hill and Park [35] in 1979 to distinguish their approach from earlier experiments where the system alternated between picture taking and moving.

Visual servoing is mainly used in cases where a robot uses a feedback machine vision in its control loop. Visual servoing is the fusion of many active research areas which includes high speed image process, kinematics, dynamics, control theory and real-time computation and has been used in control engineering especially in manufacturing fields. Visual servoing has much in common with research in active vision and structure from motion [36, 37, 38], but it is quite different from the often-described use of vision in hierarchical task level robot control systems. Some robot systems that incorporate vision are designed for task level programming, and such systems are generally hierarchical, with higher levels corresponding to more abstract data representations. The highest level is capable of reasoning about the task, given a model of the environment. Firstly, the target location and grasp sites are determined from calibrated stereo vision or laser rangefinder images, and then a sequence of moves are planned. The execution of the movements would be dealt with at the lowest level. However active vision is no more

than the use of vision at the lowest level, with simple image processing to provide active or reflexive behaviour. The active vision [39, 40] proposes that a set of simple visual behaviours can accomplish tasks through action, such as controlling attention or gaze.

On the other hand, literature related to structure from motion [41] is also relevant to visual servoing. Structure from motion attempts to infer the 3D structure and the relative motion between object and camera, from a sequence of images. In robotics however, we generally have a considerable amount of a priori knowledge of the target, and the spatial relationships between target features are known [42]. The task in visual servoing is to enable the robot to be interactive with its environment using a feedback signal in the form of an image [5]. The fundamental tenet of visual servoing is not to interpret the scene and then model it, but to direct attention to that part of the scene relevant to the task at hand. If the system wishes to learn something of the world, rather than consult the model, it should consult the world by directing the sensor. Meanwhile the benefit of an active robot-mounted camera includes the ability to avoid occlusion, resolve ambiguity and increase accuracy. Current research on visual servoing mainly focuses on the choice of image features, camera configuration, and stability of the visual servoing control system. Other topics such as rigid body pose estimation are also related to visual servoing.

2.1.2 Classification of visual servoing

In the servoing of machine vision systems, there are two methods to put the visual feedback signal into action. One is called “look and move”, and the other is visual servoing. Traditionally visual sensing and manipulation are combined in an open-loop fashion, ‘looking’ then ‘moving’. The accuracy of the operation depends directly on the accuracy of visual sensors, manipulators and their controllers. In addition, in the “look

and move" strategy, the visual information obtained by the camera or optical sensor is transformed into the position and orientation in world frame, and then guide the robot to the desired location in the world frame [5]. This method needs a priori precise calibration of cameras and manipulators. The alternative to increasing the accuracy of these subsystems is to use a visual-feedback control loop, which will increase the overall accuracy of the system. This control scheme is called visual servoing. The difference between "look and move" and "visual servoing" is that "look and move" makes use of joint feedback, whereas visual servoing uses no joint feedback information at all [43]. Hence visual servoing uses the visual information in closed-loop fashion in a control strategy to guide the robot to the desired position.

Visual servoing techniques are broadly classified into the following three types: image-based visual servoing (IBVS), position-based visual servoing (PBVS), and hybrid approach [6]. IBVS is also referred to as image feature-based technique, since it uses the image features extracted from the captured image to directly provide a command to the robot. Typically, in IBVS, all the information extracted from the image and used in control happens in 2D. The image Jacobin matrix or interaction matrix which relates displacement of image features to the motion of the camera with respect to target object is used for designing the control systems of IBVS and a closed-loop control is preformed with regard to image feature position in the image plane. Thus, a servoing control system based only on the image is constructed and should be robust against calibration errors, since there is no need to calculate the location of target objects in a 3D world frame. An important point to mention here is that the robot may still move in 3D.

PBVS is sometimes referred to as pose-based visual servoing and it is a model-based technique since the pose of the object of interest is estimated with respect to the camera and then a command is issued to the robot controller, which in turn controls the robot. In this case the image features are extracted as well. But the image feature information is used to estimate 3D information (pose of the object in 3D world frame). In PBVS, the image features are extracted from the image and are used in conjunction with other geometric information from the target to determine the pose with respect to the camera. Hence, servoing in PBVS happens in 3D.

Hybrid approaches use some combination of the 2D and 3D servoing. There have been a few different approaches to hybrid servoing, which can incorporate advantages of both IBVS and PBVS.

In summary, in IBVS, servoing is carried out on the basis of image features directly, which may reduce the computational delay, eliminate the necessity of image interpretation and eliminate errors in sensor modeling and camera calibration [44]. However it does present a significant challenge to controller design since the process is non-linear and highly coupled [43], which will cause some irregular points in the control system and make the whole control system unstable. However, in PBVS, the pose of the object is estimated, thus the control law is simple and direct. Nevertheless, because of the computational error and time delay of PBVS, the accumulation of physical error in the mechanical part of the system or even oscillation near the optimal solution point can occur [45].

2.1.3 Configuration of camera in visual servoing

In the visual servoing system, there are two basic configurations of cameras. One is to install the camera at the tip or end-effector of manipulator (eye-in-hand); the other is to set the camera and manipulator separately (eye-to-hand) [45]. According to the number of cameras used for visual servoing, it can further divided into single camera visual servoing [46], binocular camera visual servoing (two cameras) [47] and multiple cameras visual servoing.

A camera in eye-to-hand configuration can be mounted remotely with a fibre optic bundle, which is used to carry the image from the neighbourhood of the end-effector. Due to the small size and low cost of modern CCD cameras, this approach is not particularly advantageous [4]. The benefits of eye-in-hand configuration include the ability to avoid occlusion, resolve ambiguity and increase accuracy. Many reported visual servoing systems use eye-in-hand configuration, except where there is a reason that a camera cannot be mounted on the end-effector, e.g., practical constructions such as payload limitation or lack of robustness of the camera system, and an overhead camera being occluded by the gripper during the final phase of acquisition [48].

In the large-scale robotic manufacturing systems, the robot end-effector only moves in the vicinity of the desired position and the target features are chosen far away from each other [49, 50]. Therefore, the conventional configuration of cameras cannot deal with the cases mentioned above, and it is obvious that a new configuration of cameras is worthy of investigation to address this challenge.

2.1.4 3D pose estimation

In a visual servoing system, the distance between the camera and target feature is frequently referred to as depth or range [15]. The camera contains a lens that forms a 2D projection of the scene onto the image plane where the sensor is located. This projection causes the loss of direct depth information, and each point on the image plane corresponds to a ray in 3D space. Therefore, some additional information is needed to determine the 3D point corresponding to an image plane point in designing a visual servoing system. This information may come from the multiple views or knowledge of the geometric relationship between several target features. In the visual servoing system design, at least three target feature points can determine the pose of the target object with respect to the camera frame, and providing good target features is a key step. As evidenced in previous research, there are many methods to provide good target features on the scene, e.g., a pattern of light is projected on the scene. By using a single-camera vision system, a laser pointer provides a light spot to follow the planar contour [15]. Ryo Furukonwa used a laser projector, which projects a stripe of light to the measured 3D scene by hand [51]. An image feature is generally defined as any measurable relationship in an image, and examples include image moments, the relationship between regions or vertices, polygon face areas, local intensity patterns, etc. Most commonly, the coordinates of an image feature point or a region centroid are used as image features. A good image feature point is the one that can be located unambiguously in different views of the scene. After good image feature points are obtained in an image plane, there exist many approaches to get 3D space information from a 2D image plane in the visual servoing system. The main approaches relevant to visual servoing include photogrammetric

techniques [52, 53], stereo vision [54, 55], depth from motion [56], and depth from dynamics [57, 58].

Photogrammetry is the science of obtaining information about physical objects via photographic pictures. The perspective imaging model of the lens and sensor is characterized by two sets of parameters, referred to as intrinsic and extrinsic parameters. The intrinsic parameters include focal length, pixel scaling factor and the coordinate of the optical axis on the image plane. The extrinsic parameters specify the pose of the camera in the base frame. Camera calibration is the process of determining these parameters, which are generally expressed in the form of a 3×4 homogeneous transformation matrix, known as the calibration matrix. The inverse problem, camera location determination, is to find the camera pose, based on the relationship of image feature points and the intrinsic calibration parameters. The cited drawbacks of the photogrammetric approach are the complex computation, the necessity for camera calibration, and a model of the target object. The systems based on the photogrammetric principle have been demonstrated using iterative algorithms [59], Kalman filtering [60] and analytic solutions [5].

Stereo vision is the interpretation of two views of the scene taken from known different viewpoints to resolve depth ambiguity. The location of target feature points in one view must be matched with the location of the same target feature points in the other view. This matching, or correspondence problem is not trivial, and is subject to model error. Another difficulty is missing parts, where a target feature point is visible in only one of the views, therefore its depth cannot be determined [47].

Depth from motion is closely related to stereo vision. Sequential monocular views taken from different viewpoints are interpreted to derive depth information. Such a sequence may be obtained from a robot eye-in-hand camera during robot motion. It must be assumed that target features in the scene remain static during sequence acquisition [61].

As for depth from dynamics, many reported experiments utilize fixed camera-target distance. It is perhaps due to the focus or depth of field problems. The closed-loop transfer function of an image-based eye-in–hand visual servoing includes a gain term due to perspective [62, 63, 64], loop gain, and thus the closed-loop response is a function of the distance between the end-effector-mounted camera, and the target features. Conversely, the identified closed-loop dynamics can be used to derive an estimate of depth from a single point. More usefully, adaptive control or a self-tuning regulator would maintain the desired dynamic response as target distance changed, and the parameter values would be a function of target object distance.

In addition, there are a variety of vision-based depth cues including texture, perspective, stereo disparity, parallax, occlusion and shading. For a moving observer, apparent motion of features is an important depth cue. The use of multiple cues, selected according to visual circumstance, helps to resolve ambiguity. The approaches suitable for computer vision are reviewed by Jarvis [55]. Nonetheless, the estimation of depth is still a main challenge in the design of visual servoing systems.

2.1.5 Image feature

In IBVS, the geometric features of image such as points, segments or straight lines [3] are usually chosen as image features and are utilized as the inputs of visual servoing

controllers. However, these image features can only be applied in some limited target objects [7]. Also, the geometric features of the image might be occluded from the FOV of the camera. Therefore, the complete geometric image feature cannot be extracted properly. For instance, if the four corners of the image of a rectangular target object are considered as image features, one or two corner points may be covered by the intruding or unexpected object, such as the workpiece or a human hand during the operation of the system. In this case, the number of image features does not match with that of the desired image features, which may lead to the failure of visual servoing.

Recently, to track the objects that do not have enough detectable geometric features and to enhance the robustness of visual servoing systems, several novel target features are adopted for visual servoing. For example, laser points [65, 66] and the polar signature of an object contour [67] are used as target features in IBVS. Thus, the corresponding target features are used as image features for the visual servoing system.

The image moments are normally used for pattern-recognition in computer vision [23, 28, 29], and have been adopted for control scheme design due to its generic representation of any object in an image plane, with a simple or complex shape [30]. In addition, image moments can be computed easily from a binary or segmented image or from a set of extracted points of interest, disregarding the object shape complexity. Low-order moments have an intuitive meaning, since they are directly related to the area, the centroid and the orientation of the object in the image plane.

In [22], a set of image moments has been proposed as image features. Based on image moment invariants to 2-D translation, 2-D rotation, and scale, two image features are selected to control ω_x, the rotational velocity around the x axis, and ω_y, the rotational

velocity around the y axis for non-centered symmetrical objects. Using the image moments as image features in visual servoing renders the corresponding image interaction matrix with a maximal decoupled structure. Therefore, the inherent problem-singularity of the interaction matrix is avoided and hence the performance of the IBVS control system is improved. Nevertheless, these two image features cannot be used for a central symmetrical objects image plane since the elements of the interaction matrix related to these two image features are zero when the planar target object is parallel to the image plane of the camera. In this case, the last two image features S_x, S_y are proposed to control the rotational velocities ω_x , ω_y for symmetrical objects in the image plane and to avoid the singularity of the interaction matrix. However, the simulation results show that the last two image features for centered symmetrical objects used in [22] cannot represent the pose of the object correctly all the time. In this thesis, two improved image moments related to target object pose rotating around the x and y axes respectively are proposed for central symmetrical objects in the image plane. The interaction matrix describing the relationship between the motion of the camera and the variations of the proposed image features is derived, and the IBVS controller is developed based on the derived interaction matrix. The effectiveness of the proposed method is demonstrated in the experimental results on a 6-DOF robot visual servoing tracking system.

In [28], O. Tahri and F. Chaumette proposed to select the pair of image moment invariants such that the image moment error presents the global minima, with an influence zone as large and symmetrical as possible. This offline selection process has to be performed for each new object considered, once the desired image is acquired. It is noticed that in the interaction matrices of [22, 28], the rotational velocity ω_x and ω_y are

still not decoupled completely. Hence how to choose two new image features for both central symmetrical and non-symmetrical objects in the image plane to decouple the rotational velocity ω_x, ω_y and to avoid the singularity problem remains a major challenge in visual servoing. Furthermore, in previous research, the depth of target features is assumed to be constant [21] or can be calculated from the planar object equation expressed in the desired position [22, 28]. This assumption simplifies the mathematical development and such an approximate strategy is generally accurate enough in practice because of the robustness of the visual servoing scheme to modeling errors [28]. However, for large displacement robotic visual servoing tracking systems, this assumption will cause computational error and possible singularity problems with the interaction matrix. Therefore, estimating depth online can improve the accuracy of the interaction matrix and hence the visual servoing performance.

A neural network has a very strong ability for generalization and has been widely applied in function approximation and data compression, prediction, nonlinearities compensation, etc. [68, 69, 70, 71]. Recently, it is used in visual servoing systems. In [72], a new self-learning controller of robot manipulator visual servoing system with eye-in-hand configuration to track a moving object is presented, where neural networks are involved in making a direct transition from visual to joint domain without requiring calibration. Neural network approaches are also adopted to learn the relationship between the world coordinate information and the image information [73]. In addition, the neural network is trained offline and used online to provide depth information of visual servoing [74]. In this thesis, to decouple the components in the interaction matrix corresponding to ω_x, ω_y, two NN-based image features, which are proportional to the rotational angles

around the x and y axes of the camera frame respectively, are proposed. A NN is designed to estimate the rotational angles of the target object around the x and y axes of the camera frame. The interaction matrices related to the proposed image features are determined based on the estimation results of NN. In addition, the depth of the planar-shaped object is derived to further improve the computational accuracy of the overall interaction matrix and to ensure the global stability of the visual servoing system.

In summary, in visual servoing fields, the main challenges focus on three aspects. One is how to choose the configuration of cameras in visual servoing systems. The second challenge is designing visual servoing schemes. The last one is how the target features in 3D world space corresponding to image features in a 2D image plane can be obtained with some additional information. This last challenge is closely related to the determination of depth. The simplest form of visual servoing involves robot motion in a plane orthogonal to the optical axis of the camera and can be used for tracking planar motion such as a conveyor belt [75]. This visual servoing system inspires us to think that if the depth of the image features in visual servoing is known during the visual servoing process, it will be convenient to design the visual servoing controller.

2.1.6 Summary

This chapter gave the literature review on visual servoing. Additionally, the classification, history and the current state-of-the-art of visual servoing were presented. Also stated were the main problems that the researchers in this field are currently trying to solve. Lastly, three major challenges to visual servoing are identified. The subsequent chapters in this thesis are organized to address these challenges in the research area of visual servoing. Firstly, two novel multiple camera configurations of visual servoing for

large-scale robotic manufacturing are proposed in which the depths of target features are constant or can be determined online. Secondly, for the robotic tracking system, two improved image moments, which can correctly represent the pose of the target object rotating around the *x* and *y* axes of the camera frame respectively, are proposed for central symmetrical objects in image planes. In addition, to decouple ω_x, ω_y the rotational velocities around the *x* and *y* axes of the camera frames, two NN-based image features referred to as virtual image features are designed to be proportional to the rotational angles around the *x* and *y* axes of the camera frame. The depth of a planar-shaped target object can be determined online. Hence, the control performance of robotic tracking systems is improved remarkably.

CHAPTER 3 MULTIPLE CAMERA VISUAL SERVOING USED FOR LARGE-SCALE 3D POSITIONING

3.1.1 Introduction

In this chapter, a multiple camera visual servoing system is proposed for large-scale robotic manufacturing systems in the aerospace industry. The system uses eye-in-hand architecture to perform visual servoing as shown in Figure 3-1. The system consists of a 6-DOF robot, a 3D CAD laser projector, four aligned cameras, and four projection screens. The four aligned cameras [10] and four projection screens are attached to a bracket, which is fixed on the robot end-effector. A 3D CAD laser projector is installed on the robot base frame, which can project computer images and provide four target feature points on the projection screens. The cameras are used to capture the target feature points on the projection screens and obtain pose information of the workpiece in any position of the visual servoing process. This configuration ensures that all target feature points are visible for the certain limit of movement around the desired position of the robot end-effector. It is noticed that at the desired position, the laser beams remain stationary on the robot base frame and the depths of the target feature points on the projection screens, i.e., the distances between the camera centres and the projection screens, are constant during the visual servoing process. The objective of this research is to drive the workpiece to the desired position from any initial pose.

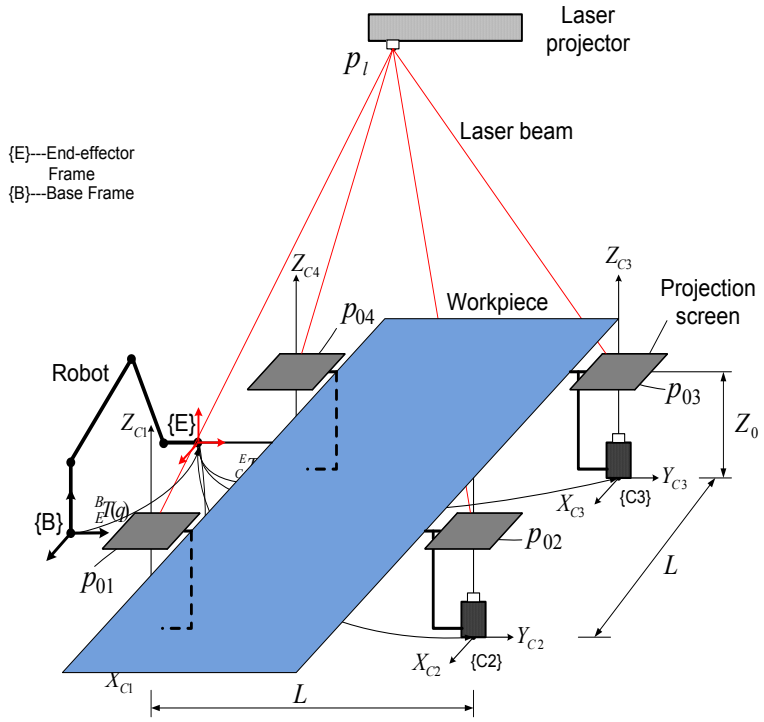


Figure 3-1 Multiple cameras with a laser projector robot visual servoing system

In Figure 3-1, the coordinate frames and transformation matrices are defined as follows: $\{B\}$ denotes robot base frame, $\{E\}$ denotes robot end-effector frame, and $\{C_i\}$ denotes the frame of camera $i\,(i=1,2,3,4)$. ${}^{B}_{E}T$ is the transformation matrix from $\{E\}$ to $\{B\}$ and ${}^{E}_{C_i}T$ is the transformation matrix from $\{C_i\}$ $(i=1,2,3,4)$ to $\{E\}$.

Visual servoing control law

3.1.2 Desired image feature coordinates

As shown in Figure 3-1, $p_l(X_l,Y_l,Z_l)$ is the coordinate of the laser projector center in the robot base frame, $p_{01}(X_{01},Y_{01},Z_{01})$, $p_{02}(X_{02},Y_{02},Z_{02})$, $p_{03}(X_{03},Y_{03},Z_{04})$ and $p_{04}(X_{04},Y_{04},Z_{04})$ are the intersection points of the laser beams and projection screens at the desired position. The image features of p_{01}, p_{02}, p_{03} and p_{04} are f_{01}, f_{02}, f_{03} and f_{04}

respectively, which are located at the centres of the image planes of the cameras and used as the desired image features at one working position (referred to in Figure 3-2). The vector of the desired image features is denoted as $f_{d0} = \begin{bmatrix} f_{01}^T & f_{02}^T & f_{03}^T & f_{04}^T \end{bmatrix}^T$.

3.1.3 Local desired image feature coordinates

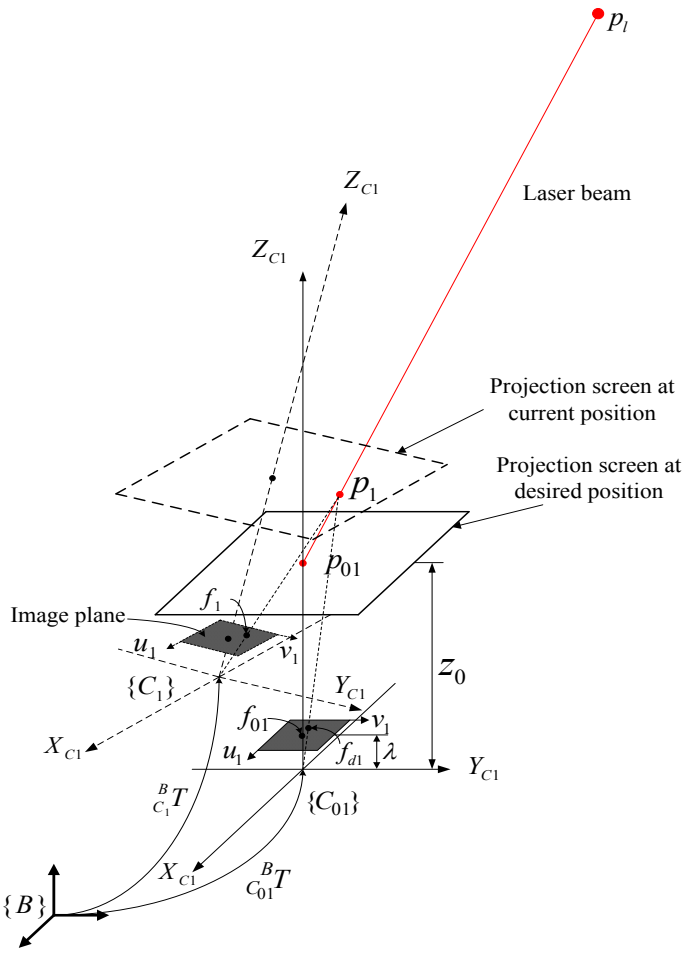


Figure 3-2 Local desired image feature

If the robot end-effector has a pose error around the desired position, the target feature points, which are the intersection points of the laser beams and the projection screens, will move along the laser beam to the new points p_1, p_2, p_3 and p_4 respectively.

Hereafter, the local desired image feature $f_{di}(x_{di}, y_{di})$ at an instant is defined as the image of p_i (i=1, 2, 3, 4) in the image plane of the camera i when locating at the desired positions. At one working position, the parametric equations of the laser beams expressed in the robot base frame can be defined as

$$\begin{cases} X = X_{0i}\xi + X_l(1-\xi) \\ Y = Y_{0i}\xi + Y_l(1-\xi) \\ Z = Z_{0i}\xi + Z_l(1-\xi) \end{cases} \qquad (i=1, 2, 3, 4)$$

where $\xi(\geq 0)$ is a parametric variable defining the relative position with regard to $p_l(X_l, Y_l, Z_l)$ and $p_{0i}(X_{0i}, Y_{0i}, Z_{0i})$. Figure 3-2 shows $f_{d1}(x_{d1}, y_{d1})$ the local desired image feature in the image plane of camera 1. The parametric equation of the laser beam $p_l p_{01}$ is rewritten as

$$\begin{cases} X = X_{01}\xi + X_l(1-\xi) \\ Y = Y_{01}\xi + Y_l(1-\xi) \\ Z = Z_{01}\xi + Z_l(1-\xi) \end{cases} \quad (\xi \geq 0) \qquad (3.1)$$

In Figure 3-2, ${}^{B}_{C_1}T$ is the transformation matrix from camera 1 frame $\{C_1\}$ to the robot base frame $\{B\}$, and ${}^{B}_{C_{01}}T$ is the transformation matrix from frame $\{C_{01}\}$ (camera 1 frame $\{C_1\}$ in the desired position) to robot base frame $\{B\}$. So the transformation matrix from robot base frame $\{B\}$ to camera 1 frame $\{C_1\}$ and to $\{C_{01}\}$ are denoted as follows

$${}^{C_1}_{B}T = {}^{B}_{C_1}T^{-1} = ({}^{B}_{E}T(q)\,{}^{E}_{C_1}T)^{-1} = {}^{E}_{C_1}T^{-1}\,{}^{B}_{E}T^{-1}(q) \qquad (3.2)$$

$${}^{C_{01}}_{B}T = {}^{B}_{C_{01}}T^{-1} = ({}^{B}_{E}T(q_0)\,{}^{E}_{C_1}T)^{-1} = {}^{E}_{C_1}T^{-1}\,{}^{B}_{E}T^{-1}(q_0) \qquad (3.3)$$

where ${}^{B}_{E}T(q)$ is the robot kinematic matrix or transformation matrix from robot end-effector frame $\{E\}$ to robot base frame $\{B\}$; q is the joint variable vector of robot and q_0 is the value of q at the desired position.

Let $\begin{bmatrix} X_{C_1} & Y_{C_1} & Z_{C_1} & 1 \end{bmatrix}^T$ be the homogenous coordinates in camera 1 frame $\{C_1\}$ of any point on laser beam $p_l p_{01}$. One obtains

$$\begin{bmatrix} X_{C_1} & Y_{C_1} & Z_{C_1} & 1 \end{bmatrix}^T = {}^{C_1}_{B}T \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T \tag{3.4}$$

Substituting (3.1) and (3.2) into (3.4) yields

$$\begin{bmatrix} X_{C_1} \\ Y_{C_1} \\ Z_{C_1} \\ 1 \end{bmatrix} = {}^{E}_{C_1}T^{-1}\,{}^{B}_{E}T^{-1}(q) \begin{bmatrix} X_{01}\xi + X_l(1-\xi) \\ Y_{01}\xi + Y_l(1-\xi) \\ Z_{01}\xi + Z_l(1-\xi) \\ 1 \end{bmatrix} \tag{3.5}$$

Hence the parametric equation of $p_l p_{01}$ expressed in camera 1 frame $\{C_1\}$ can be obtained by taking the first three equations of (3.5). It is noticed that local desired point p_1 is the intersection point of laser beam $p_l p_{01}$ and the projection screen plane. In addition, any target point on the projection screen plane in the camera 1 frame $\{C_1\}$ satisfies

$$Z_{C_1} = Z_0 \tag{3.6}$$

where Z_0 is the distance between the camera center and the projection screen plane. Substituting (3.6) into the third equation in (3.5) and solving this equation for ξ, one can obtain $\xi = \xi_1$. Thus, substituting $\xi = \xi_1$ into (3.1), one immediately obtains the coordinates of intersection point p_1 expressed in base frame $\{B\}$ as follows

$$\begin{cases} X_1 = X_{01}\xi_1 + X_l(1-\xi_1) \\ Y_1 = Y_{01}\xi_1 + Y_l(1-\xi_1) \\ Z_1 = Z_{01}\xi_1 + Z_l(1-\xi_1) \end{cases} \tag{3.7}$$

From (3.3) and (3.7), the coordinates of intersection point p_1 in camera 1 (when locating at the desired position) frame $\{C_{01}\}$ are computed as follows

$$\begin{bmatrix} X_{01} \\ Y_{01} \\ Z_{01} \\ 1 \end{bmatrix} = {}^{E}_{C_1}T^{-1}\,{}^{B}_{E}T^{-1}(q_0) \begin{bmatrix} X_{01}\xi_1 + X_l(1-\xi_1) \\ Y_{01}\xi_1 + Y_l(1-\xi_1) \\ Z_{01}\xi_1 + Z_l(1-\xi_1) \\ 1 \end{bmatrix} \tag{3.8}$$

Hence, the local desired image coordinates in the image plane of camera 1 can be readily computed as follows

$$f_{d1} = \begin{bmatrix} x_{d1} \\ y_{d1} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{Z_0} X_{01} & \frac{\lambda}{Z_0} Y_{01} \end{bmatrix}^T$$

where λ is the focal length of the camera, Z_0 is the depth of the local desired point which is equal to the distance between the camera centre and the projection screen plane. Similarly, one has

$$f_{d2} = \begin{bmatrix} x_{d2} \\ y_{d2} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{Z_0} X_{02} & \frac{\lambda}{Z_0} Y_{02} \end{bmatrix}^T, \quad f_{d3} = \begin{bmatrix} x_{d3} \\ y_{d3} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{Z_0} X_{03} & \frac{\lambda}{Z_0} Y_{03} \end{bmatrix}^T,$$

$$f_{d4} = \begin{bmatrix} x_{d4} \\ y_{d4} \end{bmatrix} = \begin{bmatrix} \frac{\lambda}{Z_0} X_{04} & \frac{\lambda}{Z_0} Y_{04} \end{bmatrix}^T$$

The vector of local desired image features is denoted as follows

$$f_{dl} = \begin{bmatrix} f_{d1}^T & f_{d2}^T & f_{d3}^T & f_{d4}^T \end{bmatrix}^T \tag{3.9}$$

3.1.4 IBVS Control law

Let $\dot{r} = \begin{bmatrix} v_{\mathrm{E}}^{\mathrm{T}} & \omega_E^T \end{bmatrix}^T = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ and

$\dot{r}_{C_i} = \begin{bmatrix} v_{Ci}^T & \omega_{Ci}^T \end{bmatrix}^T = \begin{bmatrix} v_{C_ix} & v_{C_iy} & v_{C_iz} & \omega_{C_ix} & \omega_{C_iy} & \omega_{C_iz} \end{bmatrix}^T$ be the velocity screw of the robot end-effector and the velocity screw of camera i (i=1,2,3,4) respectively. The velocity transformation matrix from $\{C_i\}$ $(i = 1,2,3,4)$ to $\{E\}$ is denoted as

$$ {}^{E}_{C_i}T_v = \begin{bmatrix} {}^{E}_{C_i}R & {}^{E}P_{C_i} \times {}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix} = \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix} \tag{3.10} $$

where ${}^{E}_{C_i}R$ is the rotational component of the transformation matrix and ${}^{E}P_{C_i}$ is the translational vector from $\{E\}$ to $\{C_i\}$, i.e.,

$$ \begin{bmatrix} v_E \\ \omega_E \end{bmatrix} = \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix} \begin{bmatrix} v_{C_i} \\ \omega_{C_i} \end{bmatrix} \tag{3.11} $$

Hence, we have

$$ \begin{bmatrix} v_{C_i} \\ \omega_{C_i} \end{bmatrix} = \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix}^{-1} \begin{bmatrix} v_E \\ \omega_E \end{bmatrix} \tag{3.12} $$

Let $f_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$, ($i$=1, 2, 3, 4) be the image features and $\dot{f}_i = \begin{bmatrix} \dot{x}_i & \dot{y}_i \end{bmatrix}^T$, ($i$=1, 2, 3, 4) be the corresponding image features' velocities. It is assumed that α_x, α_y are scaling factors of the camera and are constant in order to simplify the computation without loss of generality. The transformation between $(x_i, y_i)^T$ and the pixel indexes $(u_i, v_i)^T$ depends only on the intrinsic parameters. The following relationship between the motion of image features and the physical motion of cameras holds, i.e.,

$$\dot{f}_i = J_{image}(f_i \quad Z_o)\dot{r}_{C_i} \tag{3.13}$$

where $J_{image}(f_i \quad Z_o)$ is the interaction matrix, and Z_0 is the depth of the local desired point.

For each image feature point $(x_i \quad y_i)$, the interaction matrix is given as

$$J_{image}(f_i \quad Z_0) = \begin{bmatrix} \frac{\lambda}{Z_0} & 0 & -\frac{x_i}{Z_0} & -\frac{x_i y_i}{\lambda} & \frac{\lambda^2 + x_i^2}{\lambda} & -y_i \\ 0 & \frac{\lambda}{Z_0} & -\frac{y_i}{Z_0} & \frac{-\lambda^2 - y_i^2}{\lambda} & \frac{x_i y_i}{\lambda} & x_i \end{bmatrix} \tag{3.14}$$

Substituting (3.12) into (3.13), one obtains

$$\dot{f}_i = J_{image}(f_i \quad Z_0)\begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix}^{-1}\begin{bmatrix} v_E \\ \omega_E \end{bmatrix} = {}^{E}J_{img}(f_i \quad Z_0)\begin{bmatrix} v_E \\ \omega_E \end{bmatrix} \tag{3.15}$$

where ${}^{E}J(f_i \quad Z_0) = J_{imag}(f_i \quad Z_0)\begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix}^{-1}$.

Let $f = \begin{bmatrix} f_1^T & f_2^T & f_3^T & f_4^T \end{bmatrix}^T$ be the vector of the image features and $\dot{f} = \begin{bmatrix} \dot{f}_1^T & \dot{f}_2^T & \dot{f}_3^T & \dot{f}_4^T \end{bmatrix}^T$ be the vector of the image feature's velocity respectively. By stacking the interaction matrices of four cameras, one obtains the overall interaction matrix as follows

$$J_{image}(f \quad Z_0) = \begin{bmatrix} {}^{E}J_{image}(f_1 \quad Z_0) & {}^{E}J_{image}(f_2 \quad Z_0) & {}^{E}J_{image}(f_3 \quad Z_0) & {}^{E}J_{image}(f_4 \quad Z_0) \end{bmatrix}^T \tag{3.16}$$

Hence, the relationship between the velocities of image features and the robot end-effector's velocity screw is

$$\dot{f} = J_{image}(f \quad Z_0)\dot{r} \tag{3.17}$$

Thus the robot end-effector's velocity screw is expressed as

$$\dot{r} = J^{+}_{image}(f \quad Z_0)\dot{f} \tag{3.18}$$

where $J^{+}_{image}(f \quad Z_0)$ is the pseudo inverse interaction matrix.

If f_{dl} denotes the vector of the local desired image features, the error function is defined as $e(f) = f - f_{dl}$. We impose $\dot{e}(f) = -Ke(f)$, thus the visual servoing control law is given by

$$\dot{r} = -KJ^{+}_{imge}(f \quad Z_0)e(f) \tag{3.19}$$

where K is the proportional gain, which tunes the exponential convergence rate toward f_{dl}. The control block diagram is shown in Figure 3-3.

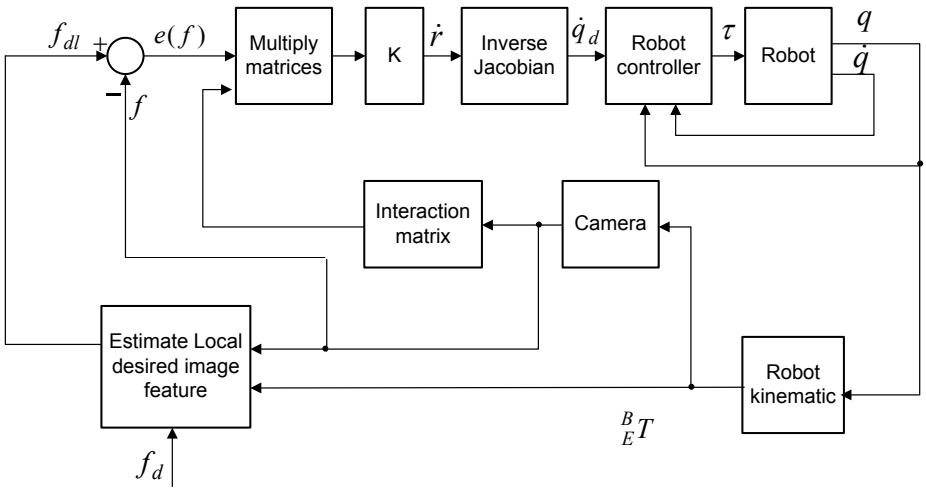


Figure 3-3 The block diagram of IBVS with multiple cameras

In Figure 3-3, $\dot{q}_d$ is the robot joint velocity in joint space and is determined by the visual servoing control law; τ is the command vector for joint torques sent by the robot controller.

3.1.5 Robot dynamic controller

It is noticed that the above-mentioned design of the visual servoing law only focuses on the kinematics of the robot without taking the dynamics of the robot and workpiece into consideration. In other words, the robot is considered as an ideal manipulation mechanism. This approach is widely adopted in most visual servoing systems where the payload of the robot is rather small. However, in the proposed large-scale robotic manufacturing systems, the size of workpiece is large and the dynamic characteristics of robot systems have a significant effect in determining performance limits and control specifications [76]. Therefore, the dynamics of robot and workpiece cannot be negligible. In this part, the dynamic model of the robot manufacturing system is considered and the robot dynamic controller is designed.

3.1.5.1 Dynamic model of the robot manufacturing system

The dynamic model of the robotic manufacturing system can be divided into two parts. One is the robot itself and the other is the workpiece plus the visual servoing system or called environment [77]. For the robot, the general form of n-joint robot dynamics in contact with the environment can be written as [78]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \Gamma_v\dot{q} + G(q) = \tau + \tau_e \tag{3.20}$$

where q is the vector of robot joint variable; $M(q)$ is a positive definite, symmetric inertia matrix; $C(q,\dot{q})$ is a vector grouping the Coriolis and centrifugal joint toques; $\Gamma_v\dot{q}$ is a vector grouping the dissipative (friction) joint torques; $G(q)$ is a vector grouping the gravity joint torques; τ is the command vector for the joint torques; τ_e is the vector of joint torques corresponding to an external wrench w_e applied by the environment to the robot.

To determine the wrench w_e applied on the robot end-effector by the environment, the gravity and inertia force acting on the workpiece are considered. To compute the inertia force acting on the workpiece, the rotational velocity and acceleration of the work piece, as well as the linear acceleration of the center of mass of the workpiece have to be known at any given instant in visual servoing. The dynamic model of the workpiece, which is held by the robot end-effector, is shown in Figure 3-4. In Figure 3-4, $\{E\}$ represents the robot end-effector frame. The up-left superscript in variables indicates that the vector is expressed in the corresponding frame, e.g., ${}^{E}\omega_e$ and ${}^{E}\dot{\omega}_e$ denote the rotational velocity and acceleration of the robot end-effector respectively expressed in frame $\{E\}$. The frame $\{C\}$ presents the workpiece frame, the centre of which is at the center of mass of the workpiece. The linear velocity and linear acceleration of the origin of frame $\{C\}$ expressed in frame $\{E\}$ is denoted as ${}^{E}v_c$ and ${}^{E}\dot{v}_c$ respectively. The transformation matrix from $\{C\}$ to $\{E\}$ is denoted as ${}^{E}_{C}T$, which is determined by the system configuration. Based on joint variable q, joint velocity $\dot{q}$, and joint acceleration $\ddot{q}$, the linear velocity

${}^{E}v_e$, linear acceleration ${}^{E}\dot{v}_e$, rotational velocity ${}^{E}\omega_e$, and rotational acceleration ${}^{E}\dot{\omega}_e$ are computed as follows [79]

$$\begin{bmatrix} {}^{E}v_e \\ {}^{E}\omega_e \end{bmatrix} = J(q)\dot{q} \tag{3.21}$$

where $J(q)$ is robot velocity Jacobian matrix; ${}^{E}v_e$ and ${}^{E}\omega_e$ are the linear velocity and rotational velocity of the robot end-effector respectively. Differentiating (3.21) with respect to time, one obtains

$$\begin{bmatrix} {}^{E}\dot{v}_e \\ {}^{E}\dot{\omega}_e \end{bmatrix} = \dot{J}(q)\dot{q} + J(q)\ddot{q} \tag{3.22}$$

Equation (3.22) is used to compute the linear acceleration ${}^{E}\dot{v}_e$ and rotational acceleration ${}^{E}\dot{\omega}_e$ of the robot end-effector in this thesis.

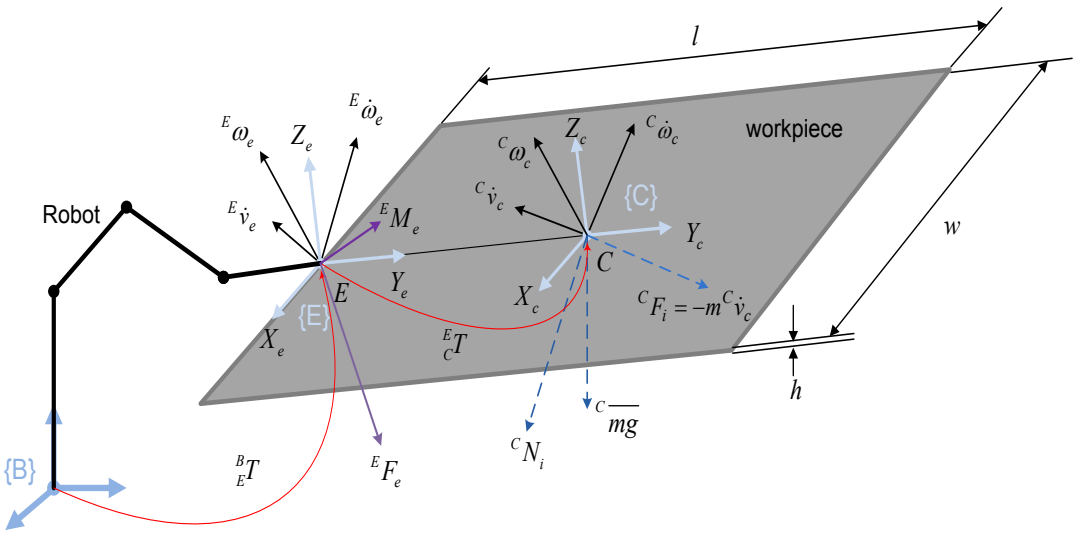


Figure 3-4 Robot and workpiece dynamic model

Hence the linear acceleration of the origin of $\{C\}$ expressed in frame $\{E\}$ [79] is expressed as

$$ {}^{E}\dot{v}_c = {}^{E}\dot{v}_e + {}^{E}\omega_e \times ({}^{E}\omega_e \times {}^{E}\overline{EC}) + {}^{E}\dot{\omega}_e \times \overline{EC} \tag{3.23} $$

where ${}^{E}\overline{EC}$ is the translational vector expressed in frame $\{E\}$.

$$ {}^{C}\dot{v}_c = {}^{C}_{E}R\,{}^{E}\dot{v}_c = ({}^{R}_{C}R)^{T}\,{}^{E}\dot{v}_e \tag{3.24} $$

where ${}^{C}\dot{v}_c$ is the linear acceleration of the origin of $\{C\}$ expressed in frame $\{C\}$, ${}^{C}_{E}R$ is the rotational matrix from frame $\{E\}$ to $\{C\}$. The robot end-effector holds the workpiece and both of them move together in the visual servoing process. Hence frames $\{E\}$ and $\{C\}$ have the same rotational velocity and rotational acceleration, i.e., ${}^{E}\omega_c = {}^{E}\omega_e$, ${}^{E}\dot{\omega}_c = {}^{E}\dot{\omega}_e$ hold.

$$ {}^{C}\omega_c = {}^{C}_{E}R\,{}^{E}\omega_c = ({}^{E}_{C}R)^{T}\,{}^{E}\omega_e \tag{3.25} $$

$$ {}^{C}\dot{\omega}_c = {}^{C}_{E}R\,{}^{E}\dot{\omega}_c = ({}^{E}_{C}R)^{T}\,{}^{E}\dot{\omega}_e \tag{3.26} $$

In such a situation, ${}^{C}F_i$ the inertia force acting at the centre of the mass of the workpiece and ${}^{C}N_i$ the inertia moment acting on the workpiece can be computed as follows

$$ {}^{C}F_i = -m\,{}^{C}\dot{v}_c \tag{3.27} $$

$$ {}^{C}N_i = -{}^{C}I\,{}^{C}\dot{\omega}_c + {}^{C}\omega_c \times {}^{C}I\,{}^{C}\omega_c \tag{3.28} $$

where m is the total mass of the workpiece and ${}^{C}I$ is the inertia tensor of the workpiece written in the frame $\{C\}$, the origin of which is located at the centre of the mass of the

workpiece. Furthermore, it is noticed that the gravity of the workpiece is a constant vector in the robot base frame $\{B\}$. Thus, the effect of the workpiece gravity on an instantaneous moment of visual servoing can be addressed quite simply by expressing the gravity vector in the frame $\{C\}$ and adding it to the inertia force vector. The gravity vector of the workpiece can be expressed in the robot base frame $\{B\}$ as ${}^{B}\overline{mg} = [0 \quad 0 \quad -mg]^{T}$, and one obtains

$${}^{C}\overline{mg} = {}^{C}_{B}R\,{}^{B}\overline{mg} = {}^{C}_{E}R\,{}^{E}_{B}R\,{}^{B}\overline{mg} = ({}^{B}_{E}R\,{}^{E}_{C}R)^{T}[0 \quad 0 \quad -mg]^{T} \tag{3.29}$$

where ${}^{C}_{E}R$ is the rotational matrix from frame $\{E\}$ to $\{C\}$, g is gravity constant. Thus the total force acting on the origin of the frame $\{C\}$, is expressed in the frame $\{C\}$ as

$${}^{C}F_{e} = {}^{C}F_{i} + {}^{C}\overline{mg}$$

The forces and torques applied at the robot end-effector or at the origin of the frame $\{E\}$ are obtained by using the force-moment transformation matrix [79] i.e.,

$$\begin{bmatrix} {}^{E}F_{e} & {}^{E}M_{e} \end{bmatrix}^{T} = {}^{E}_{C}T_{f} \begin{bmatrix} {}^{C}F_{e} & {}^{C}N_{i} \end{bmatrix}^{T} \tag{3.30}$$

where ${}^{E}_{C}T_{f} = \begin{bmatrix} {}^{E}_{C}R & 0 \\ {}^{E}\overline{EC} \times {}^{E}_{C}R & {}^{E}_{C}R \end{bmatrix}$ is the force-moment transformation matrix.

Equation (3.30) can be rewritten compactly as

$$w_{e} = {}^{E}_{C}T_{f} \begin{bmatrix} {}^{C}F_{e} & {}^{C}N_{i} \end{bmatrix}^{T} \tag{3.31}$$

The force transmission model can be obtained by applying the virtual power principle [79]. As a matter of fact, one has

$$\tau_e = J^T(q) w_e \tag{3.32}$$

where τ_e is the vector of joint torques corresponding to the external wrench w_e applied by the environment to the robot end-effector and $J^T(q)$ is the transpose of the robot velocity Jacobian matrix.

3.1.5.2 Robot controller

The structure of the robot controller is shown in Figure 3-5. In Figure 3-5, q_d, $\dot{q}_d$ and $\ddot{q}_d$ denote the desired joint angle vector, the desired rotational velocity vector and the desired rotational acceleration vector respectively. The desired rotational velocity $\dot{q}_d$ is actually generated by the visual servoing control law (3.19), i.e.,

$$\dot{q}_d = -KJ^{-1}(q) J^{+}_{image}(f \quad Z) f_e \tag{3.33}$$

where $J^{-1}(q)$ is the inverse robot velocity Jacobian matrix.

To control the complicated system like (3.20), the partitioned controller scheme [79] is adopted:

$$\tau = \alpha \tau' + \beta \tag{3.34}$$

where τ is the vector of the joint torques applied to robot joints; τ' is the vector of the joint torques representing the servo part; $\alpha = \widetilde{M}(q)$; β denotes the compensation part.

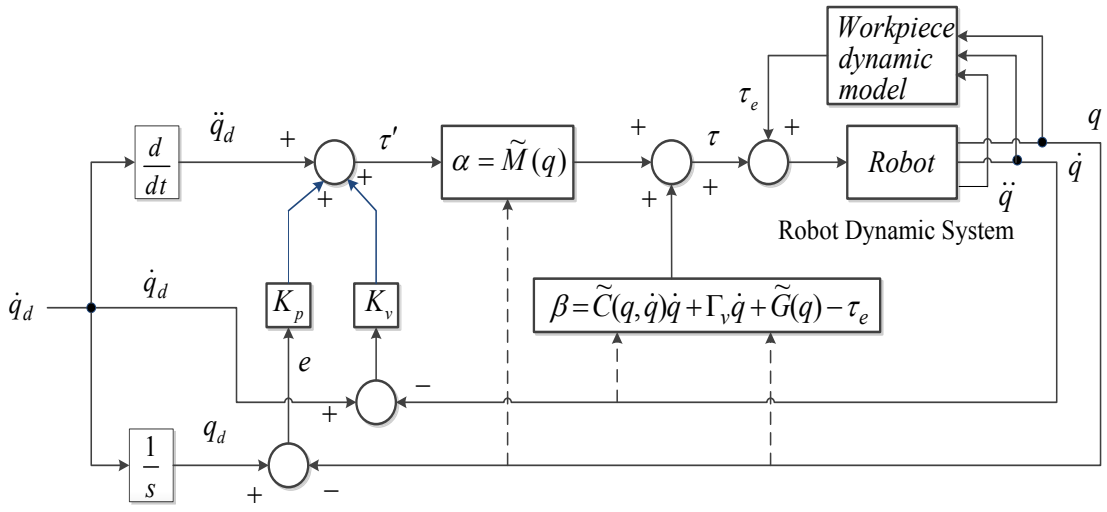


Figure 3-5 Robot controller

$$\beta = \widetilde{C}(q,\dot{q})\dot{q} + \widetilde{\Gamma}_v\dot{q} + \widetilde{G}(q) - \widetilde{\tau}_e \tag{3.35}$$

where $\widetilde{M}(q)$ is the computational inertia matrix of the robot; $\widetilde{C}(q,\dot{q})$ is a compensation for Ciriolis and centrifugal; $\widetilde{\Gamma}_v$ and $\widetilde{G}(q)$ are the compensations for the friction joint torques and the gravity respectively, $\widetilde{\tau}_e$ is the compensation to τ_e the torque applied by the robot to the environment. From the robot dynamic controller diagram shown in Figure 3-5, the compensation part is calculated as

$$\tau' = \ddot{q}_d + K_v\dot{e} + K_p e \tag{3.36}$$

The robot dynamic control law is designed as follows

$$\tau = \widetilde{M}(q)(\ddot{q}_d + K_v\dot{e} + K_p e) + \widetilde{C}(q,\dot{q})\dot{q} + \widetilde{\Gamma}_v\dot{q} + \widetilde{G}(q) - \widetilde{\tau}_e \tag{3.37}$$

where $\ddot{q}_d$ is the desired rotational acceleration vector; K_p is the proportional coefficient matrix and K_v is the differential coefficient matrix, both of which are diagonal positive

matrices; K_p is the error of the joint variable; and $\dot{e} = \dot{q}_d - \dot{q}$ is the error of the joint rotational velocity vector.

3.1.5.3 Error analysis

Substituting (3.36) and (3.37) into (3.20), if additionally, the compensation terms that are calculated by the Robotic Toolbox [80] are supposed to be precise enough, i.e., $\widetilde{M}(q) \approx M(q), \widetilde{C}(q,\dot{q}) \approx C(q,\dot{q}), \widetilde{\Gamma}_v \approx \Gamma_v$ and $\widetilde{G}_v(q) \approx G_v(q)$, one can obtain

$$\tau' = \ddot{q} - M^{-1}(q)(\widetilde{\tau}_e - \tau_e) \tag{3.38}$$

Using (3.31) and (3.33), it is quite easy to show that the closed-loop system is characterized by the error equation

$$\ddot{e} + K_v\dot{e} + K_p e = M^{-1}(q)(\widetilde{\tau}_e - \tau_e) \tag{3.39}$$

It can be inferred from the positive definiteness of K_p and K_v that $(e,\dot{e}) = (0,0)$ is the global asymptotic stable equilibrium point [79]. Furthermore, in the proposed large-scale robotic manufacturing systems, it is assumed that the robot end-effector moves in the nearby vicinity of its desired position. Hence the variation of $M(q)$ can be negligible and the equation (3.37) can be simplified by setting $M \approx M(q)$ as follows

$$\ddot{e} + K_v\dot{e} + K_p e = M^{-1}(\widetilde{\tau}_e - \tau_e) \tag{3.40}$$

where M is a constant real symmetric matrix.

Moreover, q_d , $\dot{q}_d$, $\ddot{q}_d$ are constant and the desired kinematic parameters keep updating in the iterations of visual servoing. Hence q , $\dot{q}$, $\ddot{q}$ will approach q_d , $\dot{q}_d$, $\ddot{q}_d$

respectively, which implies that the visual servoing system can reduce the image error, i.e., the robot can converge to the desired position. Meanwhile, $(\widetilde{\tau}_e - \tau_e)$ actually represents the error of the dynamic model of the robotic manufacturing system and it has a significant effect on the steady-state following-error in the iteration of visual servoing. Setting the derivatives to zero in equation (3.36) yields the steady-state equation

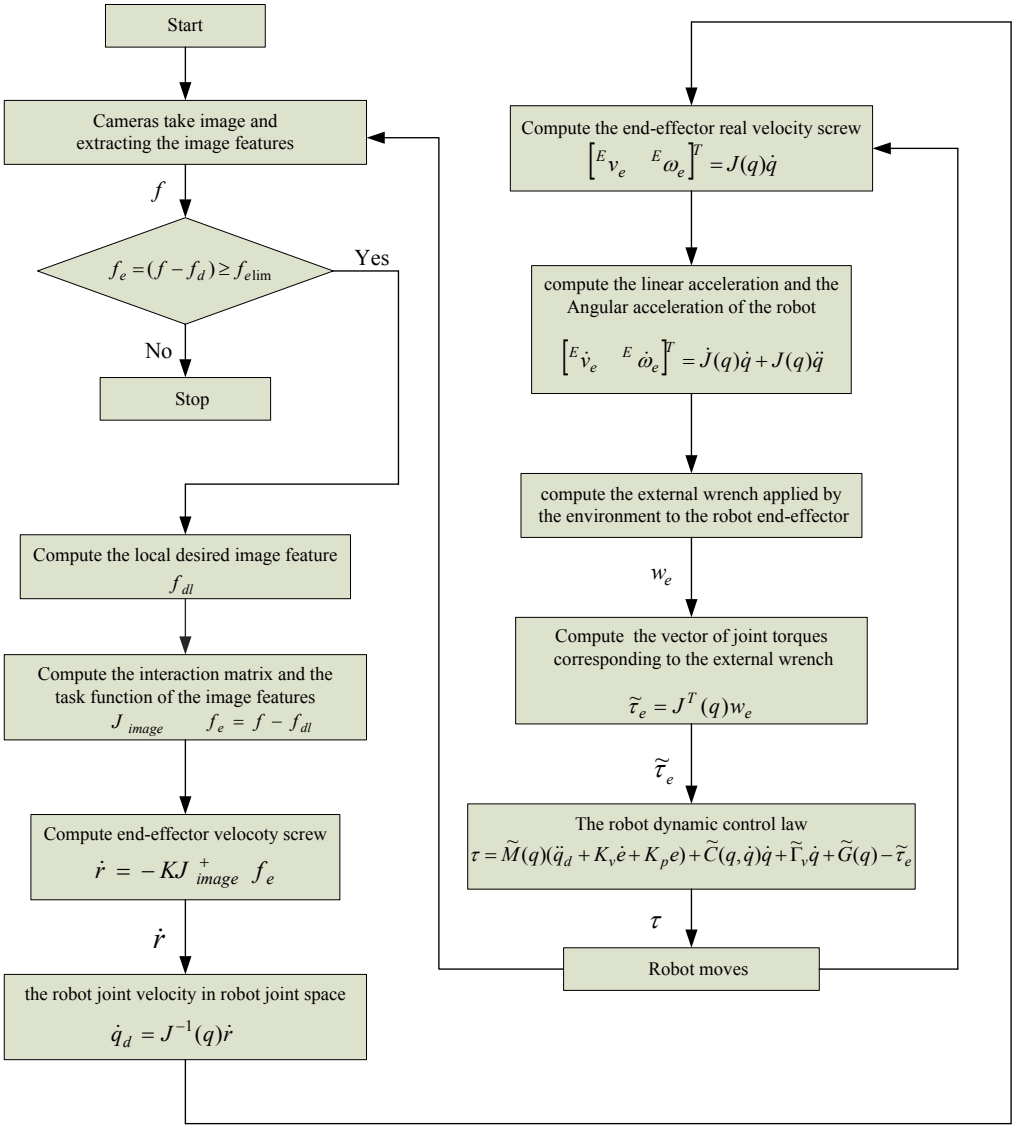
$$K_p e = M^{-1}(\widetilde{\tau}_e - \tau_e), \text{ or } e = M^{-1}(\widetilde{\tau}_e - \tau_e)/K_p \tag{3.41}$$

Thus it is clear that the smaller $(\widetilde{\tau}_e - \tau_e)$ is and the higher K_p is, the smaller the steady-state following-error will be in the iteration of visual servoing. In order to reduce the steady-state following-error and avoid oscillating of the following-error, often K_p is set to the desired closed loop stiffness and $K_v = 2K_p^{-1/2}$ for critical damping.

Form the above-analysis, it is clear that the dynamic model of robotic manufacturing system causes the joint rotational velocity following-error in visual servoing, which means that the end-effector velocity screw is different from that determined by the equation (3.17). Hence the simple proportional visual servoing control law given by (3.16) cannot be satisfied, which implies that the trajectories of the image features in image frames are not straight lines even if the proportional gain K is a scalar constant. The dynamic model error of the robotic manufacturing system or $(\widetilde{\tau}_e - \tau_e)$, may lead the trajectories of the image features in image frame to be more complex curves.

3.1.6 Control algorithm

The robot control system consists of two control loops, which are the visual servoing control loop and robot control loop. The control algorithm is summarized and shown in Figure 3-6.



where $f_{e\lim}$ is a threshold called task function precision.

Figure 3-6 The flow chart of the robot control algorithm

3.1.7 Simulation environment

In order to validate the proposed scheme, the simulations on a 6 DOF robotic manufacturing system is carried out in the Matlab/Simulink environment. The Robotic Toolbox [80] is used for robotic dynamic control implementation. In the simulations, a 6-DOF Motoman UPJ is adapted in the proposed large-scale robotic manufacturing systems. Four Sony XC55 cameras are mounted on the robot end-effector or workpiece. The focal length of the cameras λ is 6 (mm) and the scaling factor of the cameras is $\alpha_x = \alpha_y = 135135$ (pixels/m). The workpiece is a rectangular body with uniform density shown in Figure 3-4. The size of the workpiece is set as l=1.0 (m), w=1.0 (m), h=0.1 (m) and the total mass of the workpiece m=2 (kg). The coordinate axes of frame $\{C\}$ are coincident with the principle axes of the workpiece. So the inertia tensor of the workpiece written in frame $\{C\}$ is

$$ {}^{C}I = \begin{bmatrix} m(l^2+h^2)/12 & 0 & 0 \\ 0 & m(w^2+h^2)/12 & 0 \\ 0 & 0 & m(l^2+w^2)/12 \end{bmatrix} = \begin{bmatrix} 0.17 & 0 & 0 \\ 0 & 0.17 & 0 \\ 0 & 0 & 033 \end{bmatrix} $$

The coordinates (pixels) of the desired image features in the image planes of four cameras after calibration are (320, 240), (320, 240), (320, 240), and (320, 240), which are the centres of the image planes of the cameras respectively. The depths of the target feature points are set as 0.3 (m). The distance between cameras is L=1 (m). The transformation matrices of the camera frames with respect to the robot end-effector frame are represented as follows

$$ {}^{E}_{C_1}T = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{E}_{C_1}T = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 1.0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{E}_{C_3}T = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 1.0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{E}_{C_4}T = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The coordinate of the 3D CAD laser projector centre in the robot base frame is $p_l = (0,0.92,4.3)$. The coordinates of target feature points (in the desired position shown in Figure 3-1) in the robot base frame can be represented as $p_{01}(0.5,0.42,0.3)$, $p_{02}(0.5,1.42,0.3)$, $p_{03}(-0.5,1.42,0.3)$, and $p_{04}(-0.5,0.42,0.3)$. The transformation matrix from the robot end-effector frame in the desired position to the robot base frame is represented in the base frame as ${}^{B}_{E}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.42 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$. In addition, the robot control feedback matrices are set as

$$ K_p = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \end{bmatrix}, \quad K_v = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix}. $$

The desired position of the robot and workpiece is shown in Figure 3-7. In simulations, the system behavior for different kinds of movement is investigated. In case 1, the initial position of the workpiece is above the desired position. In case 2, the initial position of the workpiece is below the desired position. In both cases, the initial position errors include translational and rotational components.

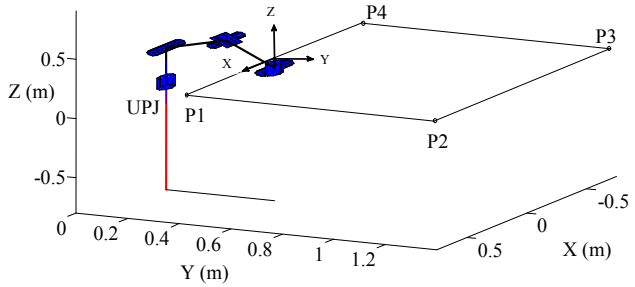


Figure 3-7 The desired position of the robot end-effector

3.1.8 Case studies

Case 1

The initial position of the robot end-effector frame in the robot base frame is represented in the transformation matrix as follow

$$ {}^{B}_{E}T = \begin{bmatrix} 0.9903 & -0.1389 & 0.0099 & -0.0370 \\ 0.1392 & 0.9842 & -0.1094 & 0.4107 \\ -0.0055 & 0.1097 & 0.9939 & 0.3121 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

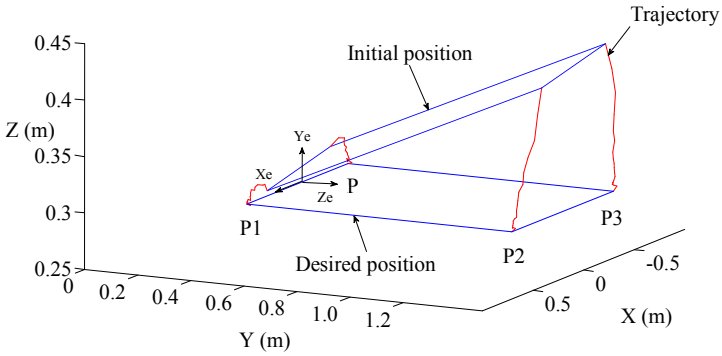


Figure 3-8 Trajectories of reference points P1, P2, P3 and P4 in the robot base frame

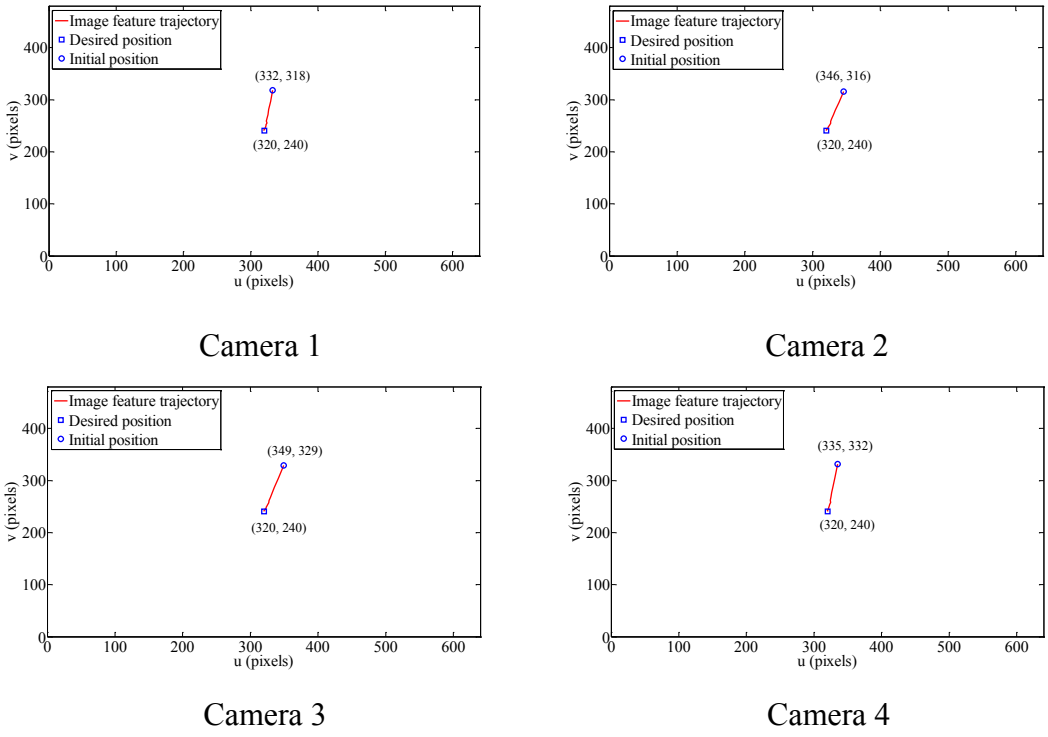


Figure 3-9 Trajectories of the image features in the image planes

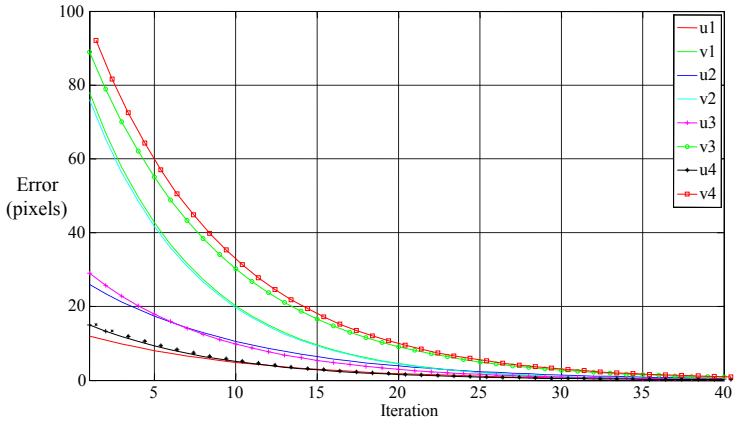


Figure 3-10 Image errors in the image plane

The trajectories of reference points on the workpiece in the robot base frame are shown in Figure 3-8, which shows that the workpiece approaches the desired position from the initial position 1 in the 3D robot base frame. Figure 3-9 shows that the trajectories of the

image features in the image planes are almost straight lines. Figure 3-10 shows that the errors of the image features converge to zero asymptotically.

Case 2

The initial position of the robot end-effector frame in the robot base frame is represented in the transformation matrix as follow:

$$ {}^{B}_{E}T = \begin{bmatrix} 0.9781 & -0.2032 & 0.0448 & -0.0762 \\ 0.2074 & 0.9691 & -0.1338 & 0.4051 \\ -0.0162 & 0.1401 & 0.9900 & 0.3179 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

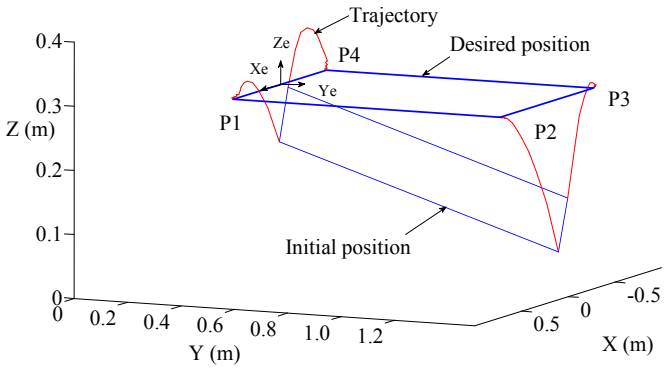
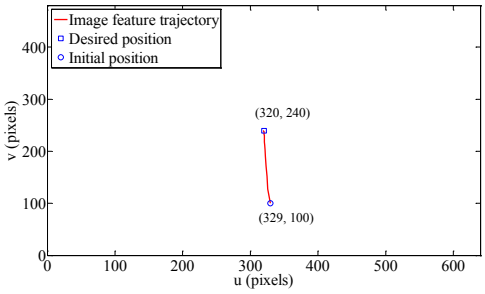
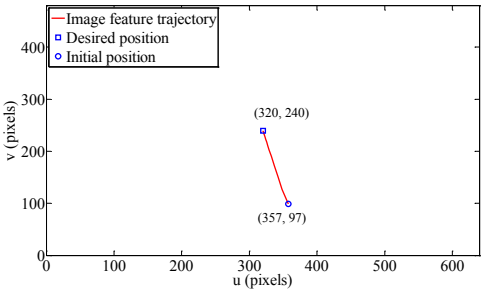


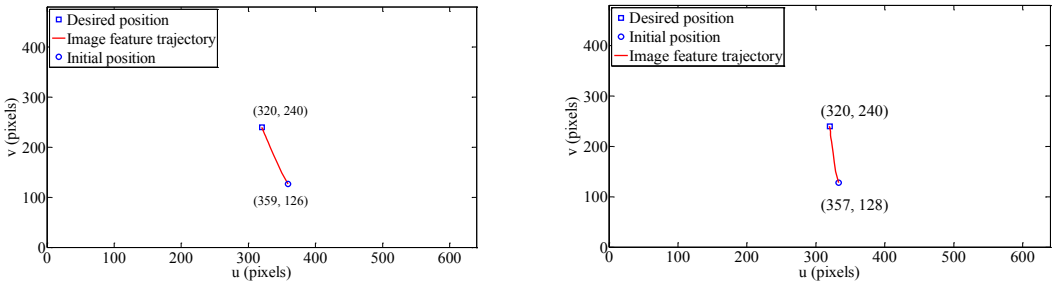
Figure 3-11 Trajectories of points P1, P2, P3 and P4 in the robot base frame



Camera 1



Camera 2



Camera 3 Camera 4

Figure 3-12 Trajectories in the image planes

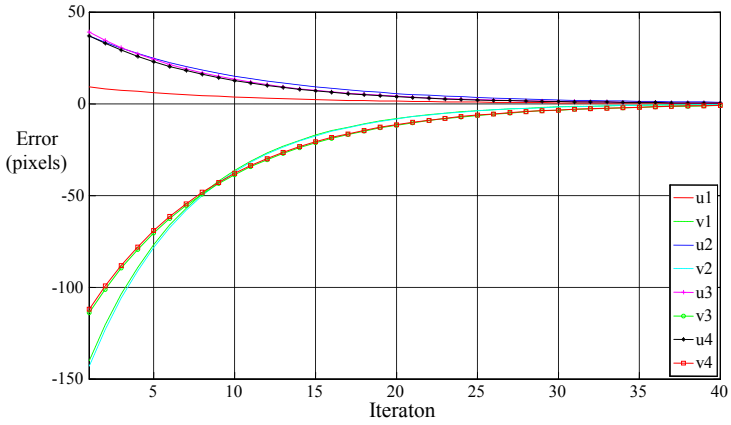


Figure 3-13 Image feature errors in the image plane

The trajectories of reference points on the workpiece in the base frame are shown in Figure 3-11, which shows that the workpiece approaches the desired position from the initial position 2 in the 3D robot base frame. Figure 3-12 shows that the trajectories of the image features in the image frame are almost straight lines. Figure 3-13 shows that the errors of image features converge to zero asymptotically.

3.1.9 Error analysis

The maximum image feature errors of four cameras is calculated as follows

$$\Delta e = \sqrt{\Delta u_{\max}{}^2 + \Delta v_{\max}{}^2}$$

In case 1, $\Delta e_1 = 0.38$ (pixels) and in case 2, $\Delta e_2 = 0.42$ (pixels).

If position error in the robot base frame is Δx, the relationship between Δx and Δe is

$$\lambda \frac{\Delta x}{Z_0} \alpha_x = \Delta e (\Delta e = \max\{\Delta e_1, \Delta e_2\})$$

The positional error in the robot base frame is

$$\Delta x = \frac{\Delta e Z_0}{\lambda \alpha_x} = 0.15 \text{ (mm)}$$

where λ, s_x, Z_0 are defined in 3.1.3 and 3.1.4.

The simulation results show that the error of multiple camera configuration of visual servoing for the large-scale robotic manufacturing system is within 0.5 (mm). It means that the proposed configuration can satisfy the need of large-scale 3D positioning in robot manufacturing systems.

Conclusion

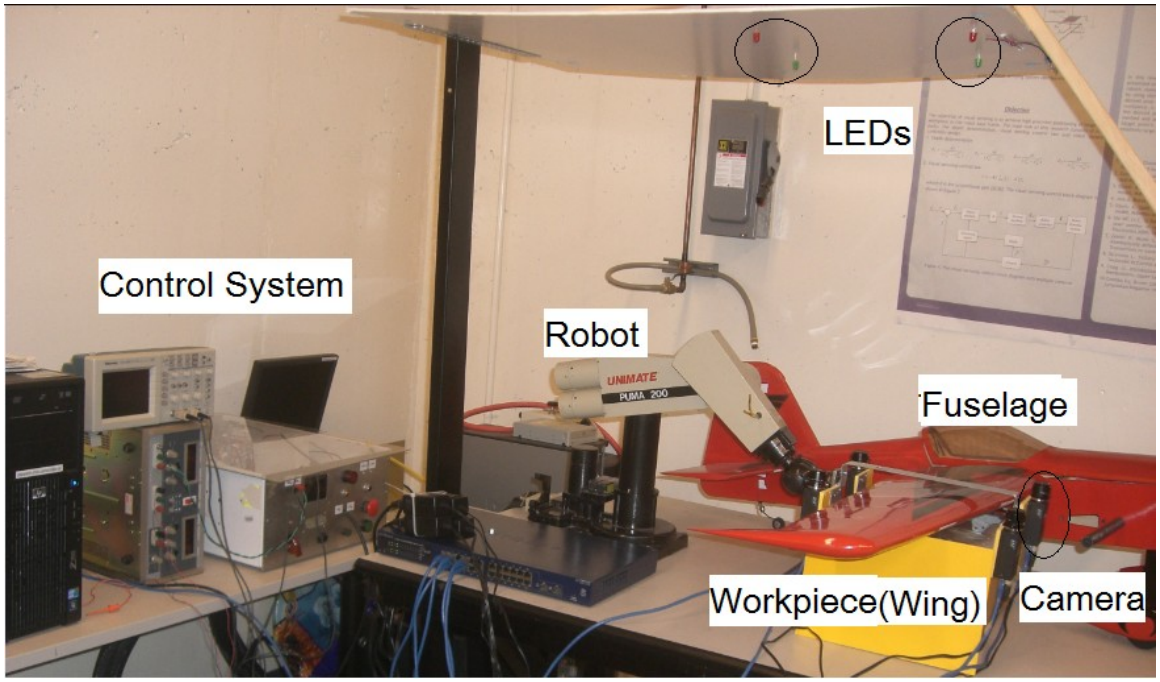
The multiple camera configuration of visual servoing for a large-scale robotic manufacturing system is proposed in this chapter. The visual servoing control law for floating target feature points and a robot dynamic controller are designed to drive the workpiece to achieve the desired pose. The simulation results verify the effectiveness of the proposed scheme and also validate the feasibility of applying the multiple camera configurations to satisfy the need of a large-scale high precision 3D positioning system in the aerospace industry.

CHAPTER 4 MULTIPLE CAMERA-MULTIPLE TARGET POINT VISUAL SERVOING IN LARGE-SCALE 3D MANUFACTURING SYSTEMS

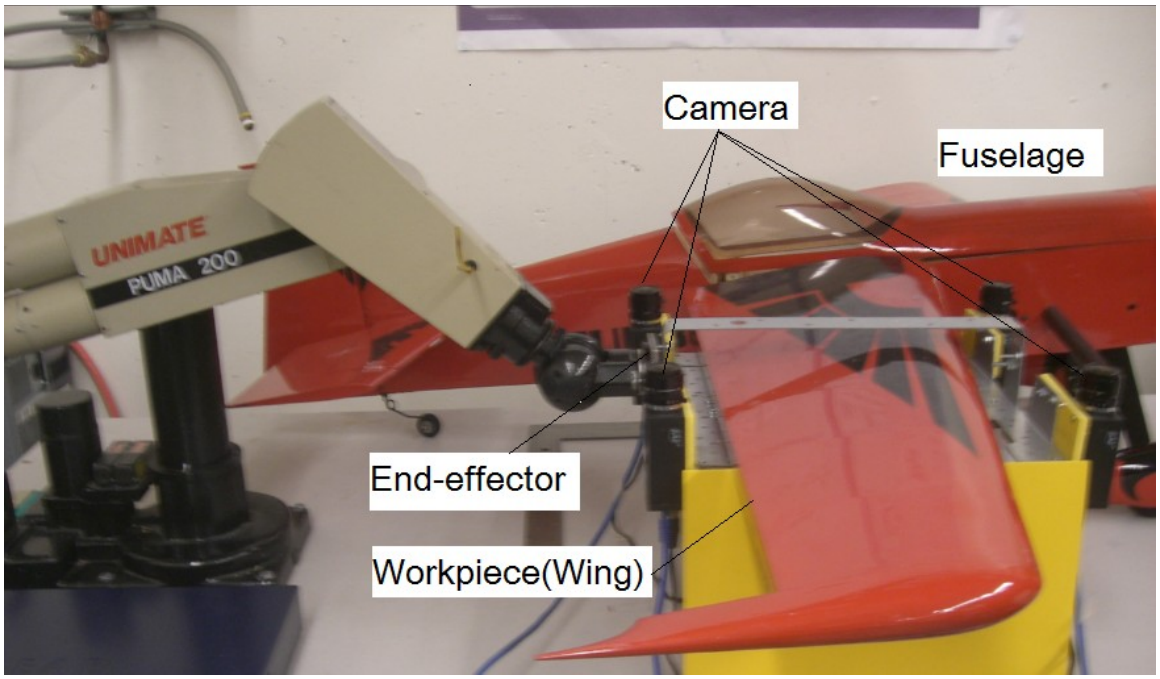
Introduction

In Chapter 3, multiple camera visual servoing with a 3D CAD laser projector for large-scale robotic manufacturing system was proposed. However, in this multiple cameras configuration of the visual servoing system, it is noticed that a costly 3D CAD laser projector is used; the laser beams are likely occluded by the objects, e.g., robot arm; and the calibration process of the system is time consuming. Hence, in this chapter, a multiple camera - multiple target point visual servoing system is proposed where the expensive laser projector is replaced by a set of LEDs. The experimental setup of the system is shown in Figure 4-1. Figure 4-1 (a) gives the system configuration, which includes a 6-DOF robot, four digital cameras, LEDs, and workpiece. Figure 4-1 (b) shows the disposal of the end-effector, cameras, and workpiece. The user interface of the multiple camera-multiple target point visual servoing system is shown in Figure 4-2.

The system adapts eye-in-hand architecture to perform visual servoing, which is shown in Figure 4-3, four principle LEDs (p_1, p_2, p_3 and p_4) are used to provide four target points on the robot base frame, and the other four complementary LEDs (q_1, q_2, q_3 and q_4) are installed in vicinity of principle LEDs which are used to determine the depths of the target feature points. Also a robot 3D pose estimation and visual servoing control system is proposed, which consists of visual servoing and robot controllers.



(a)



(b)

Figure 4-1 Multiple camera-multiple target point visual servoing system experimental setup

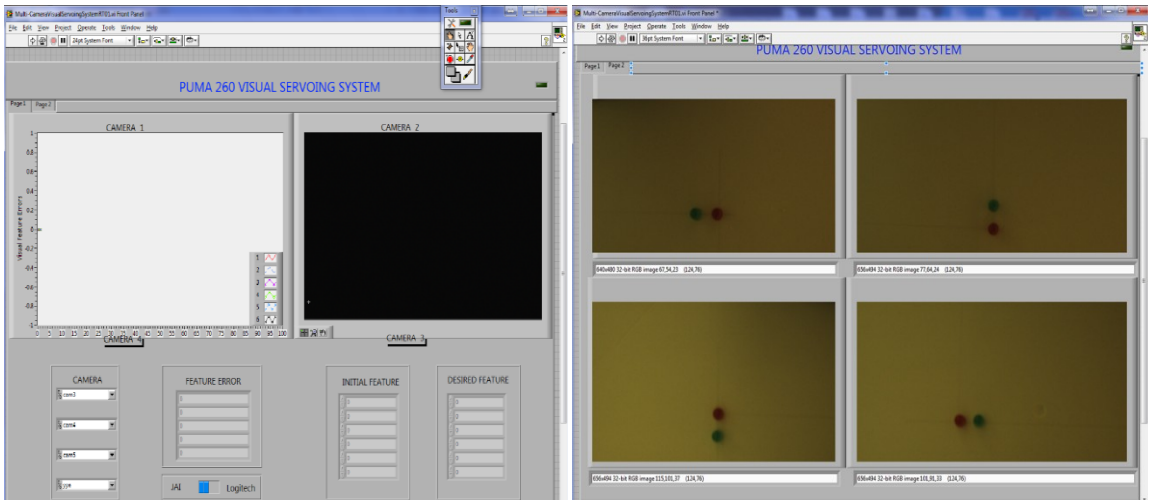


Figure 4-2 Visual servoing system user interface

A pair of LEDs (i.e., (p_1 q_1), (p_2 q_2), (p_3 q_3), (p_4 q_4)) should be in the FOV of a corresponding camera respectively in the visual servoing process. The complimentary LED is on the line linking two neighboring principle LEDs, and the distance between the principle LED and the complimentary LED in the same pair is much smaller than that between two neighboring principle LEDs (i.e., in Figure 4-3, a<<d). The objective of this research is to design a visual servoing controller to drive the workpiece to the desired position in the robot base frame from any initial pose. In Figure 4-3, the reference frames and transformation matrices are defined as follows: $\{B\}$ denotes the robot base frame, $\{E\}$ denotes the robot end-effector frame, and $\{C_i\}$ denotes the frame of camera i $(i = 1,2,3,4)$. ${}^{B}_{E}T$ is the transformation matrix from $\{E\}$ to $\{B\}$, and ${}^{E}_{C_i}T$ is the transformation matrix from $\{C_i\}$ $(i = 1,2,3,4)$ to $\{E\}$.

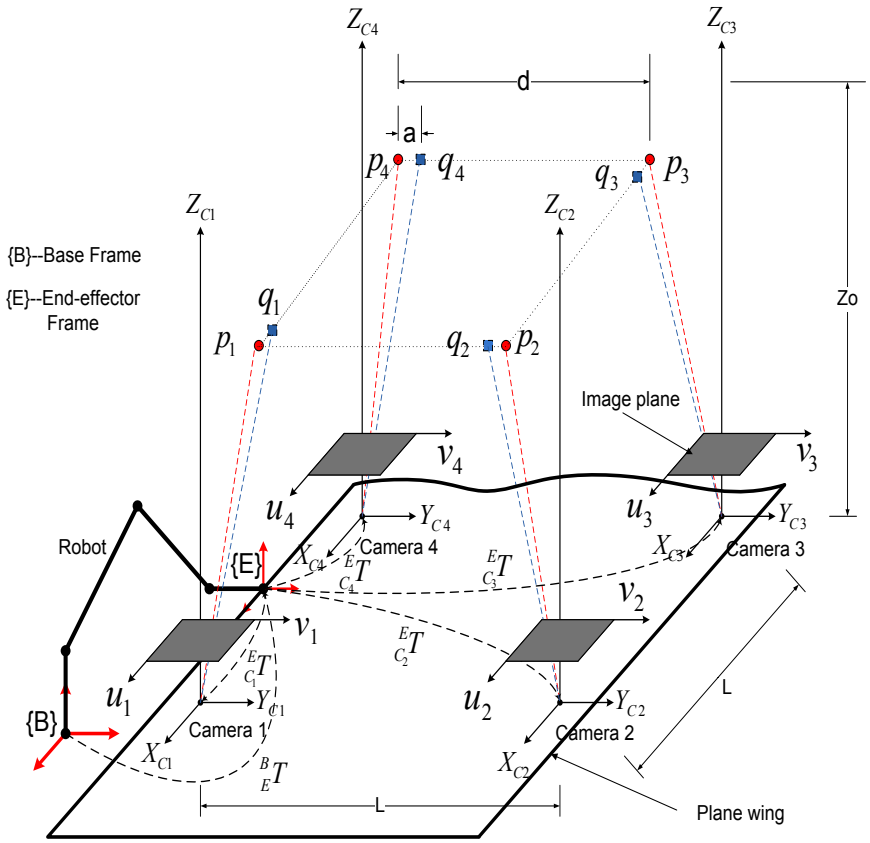


Figure 4-3 Multiple cameras-multiple target point visual servoing system

Depth computation

It is well-known that the interaction matrix in IBVS involves the depths of target feature points [28]. Hence, obtaining the accurate depths of target feature points online remains one of the challenges for IBVS. In this part, an online depth estimation method is proposed and a pose estimation algorithm for multiple camera-multiple target point visual servoing system for large scale robotic manufacturing systems is developed.

In the proposed configuration, two neighboring cameras are used to determine the depth of one target feature point, for instance camera 1 and 4 are used to determine the depth of target feature point p_4 with respect to camera 4, which is shown in Figure 4-3. If a target feature point is visible to both cameras simultaneously, it is easy to determine

the depth by using the stereo vision method. However, in the proposed configuration, because the four cameras are far away from each other, one target feature point can only be observed by one camera, i.e., p_1 by camera 1, p_2 by camera 2, and so on. In order to determine the depth of a target feature point by stereo vision, one must know the coordinates of the image feature point in two camera image planes. As mentioned above, one camera cannot capture two feature points in its FOV, e.g., camera 1 cannot take target feature points p_1 and p_4 into its FOV at the same time. To address this challenge, the complementary feature point q_1, q_2, q_3 and q_4 are added into the system shown in Figure 4-3. It is noticed that q_1, q_2, q_3, and q_4 are placed in the vicinity of feature point p_1, p_2, p_3, and p_4 and located on the lines of p_1p_4, p_2p_1, p_3p_2, and p_4p_3 respectively.

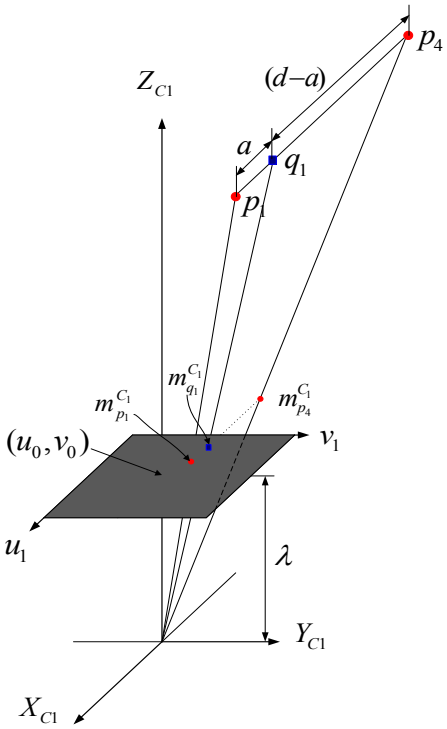


Figure 4-4 Illustration of the virtual image coordinates of p_4 in the image plane of camera 1

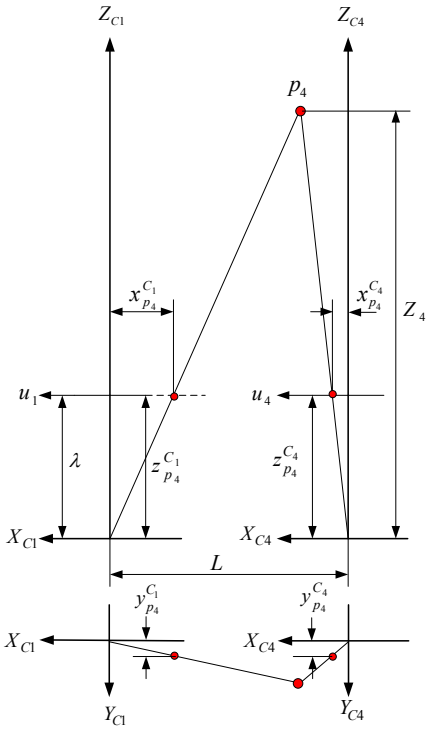


Figure 4-5 Depth computation of target point p_4

In order to estimate the depth Z_4 of feature point p_4 with respect to camera 4, camera 1 and 4 are used as one pair. Similarly, camera 3 and 4 are used to determine the depth Z_3 of feature point p_3 with respect to camera 3, and so on. In Figure 4-4, the depth Z_4 of feature point p_4 with respect to camera 4 is taken as an example to demonstrate the procedure of depth computation. In the proposed configuration, since feature point p_4 is out of the FOV of camera 1 in the visual servoing process, the coordinates of the image feature of p_4 in the image plane of camera 1 cannot be observed directly by camera 1. Therefore, complimentary feature point q_1 is added in the vicinity of target feature point p_1, both of which are within the FOV of camera 1 in the visual servoing process. It is assumed that $m_{p_1}^{C_1}(u_{p_1}^{C_1}, v_{p1}^{C_1})$ and $m_{q_1}^{C_1}(u_{q_1}^{C_1}, v_{q_1}^{C_1})$ denote the image features of the target feature point p_1 and complimentary feature point q_1 in the image plane of camera 1 respectively. The complimentary feature point q_1 is located on the line of p_1p_4 and the distance between p_1 and q_1 is denoted as a which is pre-defined by the system setup. If the image plane of camera 1 could extend large enough, which is called the virtual image plane, the image coordinates of the image feature of feature point p_4 in the virtual image plane of camera 1 can be denoted as $m_{p_4}^{C_1}(u_{p_4}^{C_1}, v_{p_4}^{C_1})$. According to the computer vision [81], one has

$$\frac{(u_{q1}^{C_1} - u_{p4}^{C_1})}{(u_{p1}^{C_1} - u_{q1}^{C_1})} = \frac{(d-a)}{a} \tag{4.1}$$

$$\frac{(v_{q1}^{C_1} - v_{p4}^{C_1})}{(v_{p1}^{C_1} - v_{q1}^{C_1})} = \frac{(d-a)}{a} \tag{4.2}$$

From (4.1) and (4.2), one has $u_{p_4}^{C_1} = u_{q_1}^{C_1} - \frac{(d-a)}{a}(u_{p_1}^{C_1} - u_{q_1}^{C_1})$ and $v_{p_4}^{C_1} = v_{q_1}^{C_1} - \frac{(d-a)}{a}(v_{p_1}^{C_1} - v_{q_1}^{C_1})$.

As shown in Figure 4-5, the coordinates of $m_{p_4}^{C_1}$ in $\{C_1\}$ the frame of camera 1, i.e., $(x_{p_4}^{C_1}, y_{p_4}^{C_1}, z_{p_4}^{C_1})$ are calculated as

$$x_{p_4}^{C_1} = \alpha_x(u_{p_4}^{C_1} - u_0) \tag{4.3}$$

$$y_{p_4}^{C_1} = \alpha_x(v_{p_4}^{C_1} - v_0) \tag{4.4}$$

$$z_{p_4}^{C_1} = \lambda \tag{4.5}$$

where λ is the camera focal length, α_x, α_y are scaling factors of the camera and (u_0, v_0) are the coordinates of the image center point in the image frame. In the proposed configuration, four cameras in the system have the same intrinsic parameters. If the image feature of target feature point p_4 in the image plane of camera 4 is denoted as $m_{p_4}^{C_4}(u_{p_4}^{C_4}, v_{p_4}^{C_4})$, and the coordinates of $m_{p_4}^{C_4}$ in the frame of camera 4 are denoted as $(x_{p_4}^{C_4}, y_{p_4}^{C_4}, z_{p_4}^{C_4})$, one has

$$x_{p_4}^{C_4} = \alpha_x(u_{p_4}^{C_4} - u_0) \tag{4.6}$$

$$y_{p_4}^{C_4} = \alpha_y(v_{p_4}^{C_4} - v_0) \tag{4.7}$$

$$z_{p_4}^{C_4} = \lambda \tag{4.8}$$

From the coordinate plane $X_{C1} - Z_{C1}$ or $X_{C4} - Z_{C4}$, as shown in Figure 4-5, one obtains [2]

$$\frac{(Z_4 - \lambda)}{Z_4} = \frac{L - (x_{p4}^{C1} - x_{p4}^{C4})}{L} \tag{4.9}$$

where Z_4 is the depth of target feature point p_4 with respect to camera 4, and L is the distance between camera 1 and camera 4. From equations (4.5), (4.7) and (4.9), the depth of feature point p_4 with respect to camera 4 is calculated as

$$Z_4 = \frac{\lambda L}{(x_{p_4}^{C_1} - x_{p_4}^{C_4})} \tag{4.10}$$

Similarly, one has

$$Z_3 = \frac{\lambda L}{(y_{p_3}^{C_4} - y_{p_3}^{C_3})} \tag{4.11}$$

$$Z_2 = \frac{\lambda L}{(x_{p_2}^{C_3} - x_{p_2}^{C_2})} \tag{4.12}$$

$$Z_1 = \frac{\lambda L}{(y_{p_1}^{C_2} - y_{p_1}^{C_1})} \tag{4.13}$$

Equations (4.10) to (4.13) are used to compute the depths of feature points online during visual servoing.

Visual servoing control law

The objective of visual servoing is to design the visual servoing controller to drive the robot to achieve high precision positioning of the large-scale workpiece. Let

$\dot{r} = \begin{bmatrix} {}^E v_{\mathrm{e}}^{\mathrm{T}} & {}^E \omega_e^T \end{bmatrix}^T = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ be the velocity screw of the robot end-effector expressed in the robot end-effector frame $\{E\}$ and

$\dot{r}_{C_i} = \begin{bmatrix} v_{Ci}^T & \omega_{Ci}^T \end{bmatrix} = \begin{bmatrix} v_{C_i x} & v_{C_i y} & v_{C_i z} & \omega_{C_i x} & \omega_{C_i y} & \omega_{C_i z} \end{bmatrix}^T$ be the velocity screw of the camera i (i=1, 2, 3, 4) expressed in the corresponding camera frame $\{C_i\}$ (i=1, 2, 3, 4). The velocity transformation matrix from the camera frame $\{C_i\}$ (i=1, 2, 3, 4) to the robot end-effector frame $\{E\}$ is denoted as

$$ {}^{E}_{C_i}T_v = \begin{bmatrix} {}^{E}_{C_i}R & {}^{E}P_{C_i} \times {}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix} = \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix} $$

where ${}^{E}_{C_i}R$ is the rotational component of transformation matrix ${}^{E}_{C_i}T_v$ and ${}^{E}P_{C_i}$ is the translational vector from $\{E\}$ to $\{C_i\}$ (i=1,2,3,4), i.e.,

$$ \begin{bmatrix} {}^{E}v_e \\ {}^{E}\omega_e \end{bmatrix} = \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix} \begin{bmatrix} v_{C_i} \\ \omega_{C_i} \end{bmatrix} \tag{4.14} $$

Hence, one has

$$ \begin{bmatrix} v_{C_i} \\ \omega_{C_i} \end{bmatrix} = \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix}^{-1} \begin{bmatrix} {}^{E}v_e \\ {}^{E}\omega_e \end{bmatrix} \tag{4.15} $$

It is assumed that $f_i = [x_i \quad y_i]^T$ (i=1, 2, 3, 4) represents the image features; $\dot{f}_i = [\dot{x}_i \quad \dot{y}_i]^T$ (i=1, 2, 3, 4) represents the corresponding image feature velocities; $f_{di} = [x_{di} \quad y_{di}]^T$ (i=1, 2, 3, 4) is the desired image features; $f_d = [f_{d1}^T \quad f_{d2}^T \quad f_{d3}^T \quad f_{d4}^T]^T$ is the vector of the desired image features, which can be obtained through system calibration. It is well known that α_x, α_y the scaling factors of the camera are constant, the transformation between $[x_i \quad y_i]^T$ and $[u_i \quad v_i]^T$ the pixel indexes depends only on the intrinsic parameters. It is well known that the following relationship between the motion of the image features and the physical motion the camera holds

$$ \dot{f}_i = J_{image}(f_i \quad Z_i)\dot{r}_{C_i} \tag{4.16} $$

where $J_{image}(f_i \quad Z_i)$ is the interaction matrix. For each feature point $(x_i \quad y_i)$, the interaction matrix is computed as follows

$$J_{image}(f_i \quad Z_i) = \begin{bmatrix} \frac{\lambda}{Z_i} & 0 & -\frac{x_i}{Z_i} & -\frac{x_i y_i}{\lambda} & \frac{\lambda^2 + x_i^2}{\lambda} & -y_i \\ 0 & \frac{\lambda}{Z_i} & -\frac{y_i}{\lambda} & \frac{-\lambda^2 - y_i^2}{\lambda} & \frac{x_i y_i}{\lambda} & x_i \end{bmatrix} \tag{4.17}$$

Substituting (4.13) into (4.14), one obtains

$$\dot{f}_i = J_{image}(f_i \quad Z_i) \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix}^{-1} \begin{bmatrix} {}^{E}v_e \\ {}^{E}\omega_e \end{bmatrix} = {}^{E}J_{image}(f_i \quad Z_i) \begin{bmatrix} {}^{E}v_e \\ {}^{E}\omega_e \end{bmatrix} \tag{4.18}$$

where ${}^{E}J_{imae}(f_i \quad Z_i) = J_{image}(f_i \quad Z_i) \begin{bmatrix} {}^{E}_{C_i}R & sk({}^{E}P_{C_i}){}^{E}_{C_i}R \\ 0 & {}^{E}_{C_i}R \end{bmatrix}^{-1}$.

Let $Z = [Z_1 \quad Z_2 \quad Z_3 \quad Z_4]^T$ be the vector of the depths of feature points, $f = [f_1^T \quad f_2^T \quad f_3^T \quad f_4^T]^T$ be the vector of four image feature coordinates, and $\dot{f} = [\dot{f}_1^T \quad \dot{f}_2^T \quad \dot{f}_3^T \quad \dot{f}_4^T]^T$ be the vector of four image feature velocities. By stacking the interaction matrices of four image features, one obtains the overall interaction as follows

$$J_{image}(f \quad Z) = [{}^{E}J_{image}(f_1 \quad Z_1) \quad {}^{E}J_{image}(f_2 \quad Z_2) \quad {}^{E}J_{image}(f_3 \quad Z_3) \quad {}^{E}J_{image}(f_4 \quad Z_4)]^T \tag{4.19}$$

Hence, the relationship between the motion of the image features and the robot end-effector velocity screw is denoted as

$$\dot{f} = J_{img}(f \quad Z)\dot{r}$$

The end-effector velocity screw can be expressed as

$$\dot{r} = J^{+}_{image}(f \quad Z)\dot{f}$$

where $J^{+}_{image}(f \quad Z)$ is the pseudo inverse of the interaction matrix. If the task function of the image features is defined as $f_e = f - f_d$ and we impose a simple proportional control law

$$\dot{f}_e = -Kf_e \tag{4.20}$$

Thus the visual servoing control signal is

$$\dot{r} = -KJ^{+}_{image}(f \quad Z)f_e \tag{4.21}$$

where $\dot{r}$ is the end-effector velocity screw sent to the robot controller, K is the proportional gain which tunes the exponential convergence rate toward f_d.

The desired joint velocity in the robot joint space, which is sent to the robot controller, is obtained as follows

$$\dot{q}_d = J^{-1}(q)\dot{r} \tag{4.22}$$

where $J(q)$ is the forward kinematic matrix of the robot.

The block diagram of the visual servoing system

The block diagram of the visual servoing control system with multiple cameras is shown in Figure 4-6, which includes the visual servoing control loop and the robot dynamic control loop.

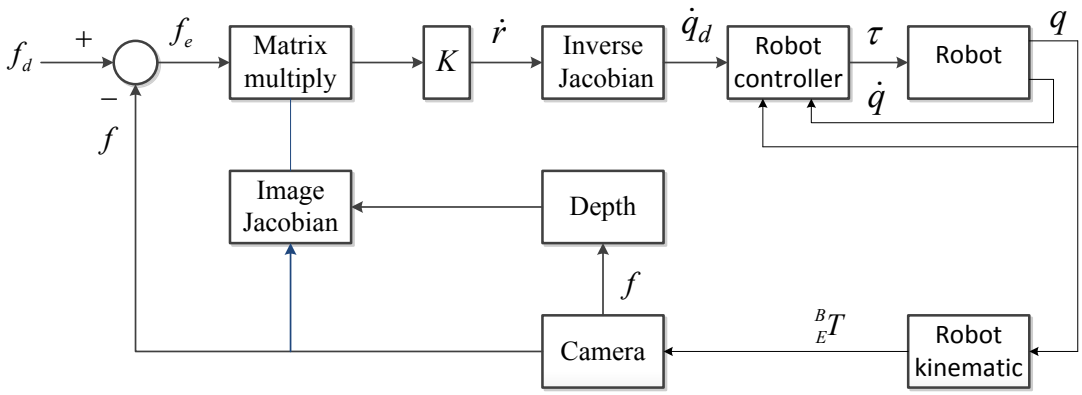
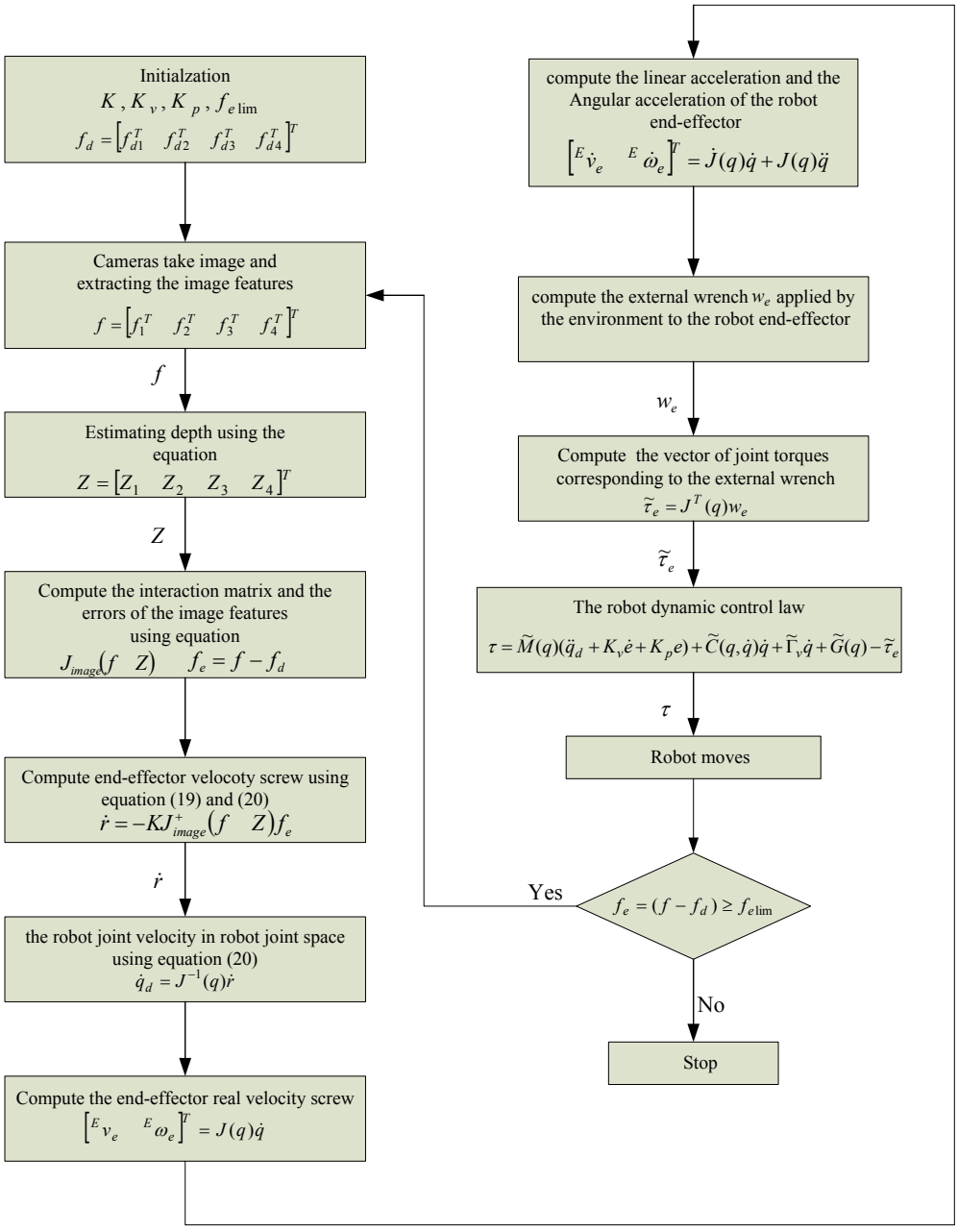


Figure 4-6 Block diagram of the visual servoing control system with multiple cameras

In Figure 4-6, the $\dot{q}_d$ is the robot joint velocity in the robot joint space, which is determined by the visual servoing control law (4.22) and τ is the command vector of the joint torques sent by the robot dynamic controller. The design of the robot dynamic controller is referred to in CHAPTER 3.

Visual servoing control algorithm

The control algorithm is summarized in Figure 4-7.



where $f_{e\lim}$ is a threshold called task function precision.

Figure 4-7 Flow chart of the robot control algorithm

4.1.1 Simulation environment

In order to validate the proposed scheme, the simulation on a 6 DOF robotic manufacturing system has been carried out in the Matlab/Simulink environment. The Robotic Toolbox [80] is used for robotic dynamic control implementation. In the simulation, a 6-DOF Motoman UPJ is used as the robot in the proposed large-scale manufacturing systems. Four Sony XC55 cameras are mounted on the robot end-effector or workpiece. The focal length of the cameras is $\lambda = 6$ (mm), and the scaling factor of the cameras is $\alpha_x = \alpha_y = 135135$ (pixels/m). Four principle LEDs (p_1, p_2, p_3 and p_4) provide four target feature points for visual servoing; and the other four complementary LEDs (q_1, q_2, q_3 and q_4) are used to determine the depths of target feature points online. All the LEDs are installed in the robot base frame shown in Figure 4-3. The geometric parameters are set as $a = 0.1$ (m), $d = 0.9$ (m), and $L = 1$ (m). The workpiece is a rectangular body of uniform density shown in Figure 3-4. The size of the workpiece is set as $l = 1.4$ (m). $w = 1$ (m) $h = 0.1$ (m), and the total mass of the workpiece $m = 2.5$ (kg). The coordinate axes of frame $\{C\}$ are coincident with the principle axes of the workpiece. The inertia tensor of the workpiece written in frame $\{C\}$ is

$$ {}^{C}I = \begin{bmatrix} m(l^2+h^2)/12 & 0 & 0 \\ 0 & m(w^2+h^2)/12 & 0 \\ 0 & 0 & m(l^2+w^2)/12 \end{bmatrix} = \begin{bmatrix} 0.41 & 0 & 0 \\ 0 & 0.21 & 0 \\ 0 & 0 & 0.62 \end{bmatrix} $$

The desired coordinates (pixels) of the image feature points in four camera image frames are (302.37, 257.63), (302.37, 222.37), (337.63, 222.37), and (337.63, 257.63) respectively. The initial depth Z_0 is 2.3 (m). The error of the camera model is 5%. The

transformation matrices of the camera frames with respect to the robot end-effector frame are represented as follows

$$ {}^{E}_{C_1}T = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{E}_{C_2}T = \begin{bmatrix} 1 & 0 & 0 & 0.5 \\ 0 & 1 & 0 & 1.0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{E}_{C_3}T = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 1.0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad {}^{E}_{C_4}T = \begin{bmatrix} 1 & 0 & 0 & -0.5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The transformation matrix of the robot end-effector frame in the desired position with respect to the robot base frame is represented as follows

$$ {}^{B}_{E}T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0.42 \\ 0 & 0 & 1 & 0.3 \\ 0 & 0 & 0 & 1 \end{bmatrix}. $$

Thus the coordinates of the target feature points in the robot base frame can be represented as follows: $p_1(0.45,0.47,2.3)$, $p_2(0.45,1.37,2.3)$, $p_3(-0.45,1.37,2.3)$, and $p_4(-0.45,0.47,2.3)$. The desired position of the workpiece is shown in Figure 4-8. In addition, the proportional gain is set as $K = 0.1$; the robot control feedback matrices are set as

$$ K_p = \begin{bmatrix} 9 & 0 & 0 & 0 & 0 & 0 \\ 0 & 9 & 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 0 & 9 & 0 \\ 0 & 0 & 0 & 0 & 0 & 9 \end{bmatrix}, \quad K_v = \begin{bmatrix} 6 & 0 & 0 & 0 & 0 & 0 \\ 0 & 6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 6 \end{bmatrix}. $$ The feature error threshold $f_{e\lim}$ is set as 0.5 pixel in simulation studies.

4.1.2 Case studies

The system behavior for different kinds of movement is investigated. In case 1, the initial position of the workpiece is above the desired one of the workpiece and in case 2, the initial position of the workpiece is below the desired one of the workpiece. In both cases, the initial position error includes translational and rotational components.

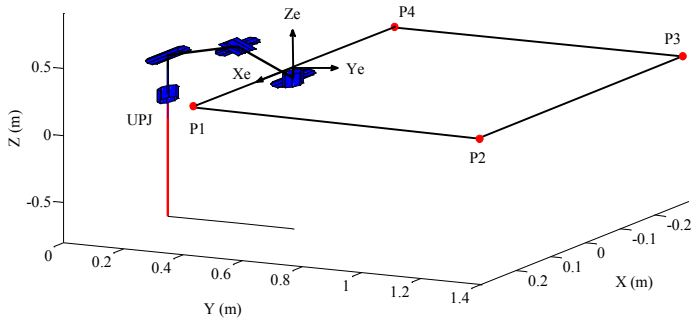


Figure 4-8 Desired position of the robot end-effector

Case 1

The initial position of the robot end-effector frame in the robot base frame is represented in the transformational matrix as follows

$$ {}^{B}_{E}T = \begin{bmatrix} 0.9903 & -0.1389 & 0.0099 & -0.0370 \\ 0.1392 & 0.9842 & -0.1094 & 0.4107 \\ -0.0055 & 0.1097 & 0.9939 & 0.3121 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The control objective is to drive the workpiece to the desired position from the initial positions. The simulation results are shown in Figure 4-9 to Figure 4-11.

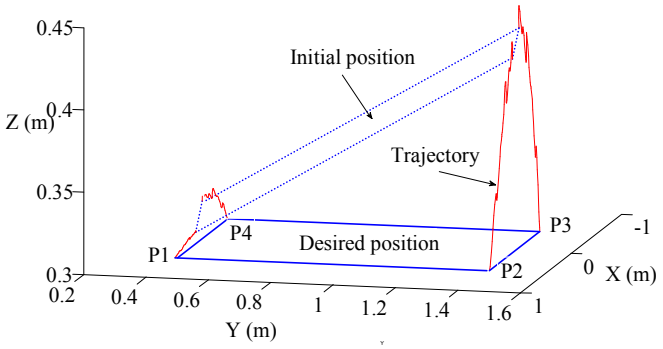
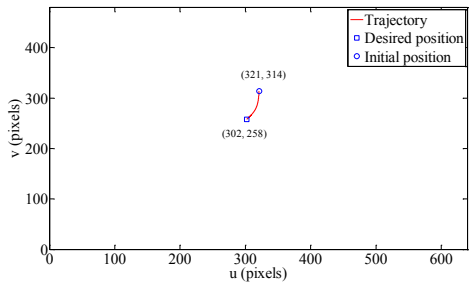
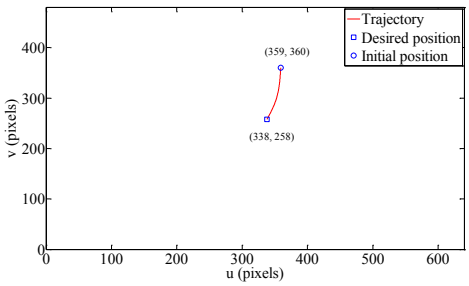


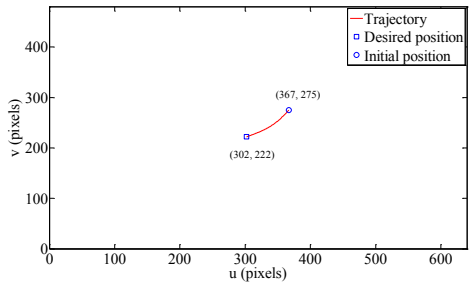
Figure 4-9 Trajectories of points P1, P2, P3 and P4 in the robot base frame



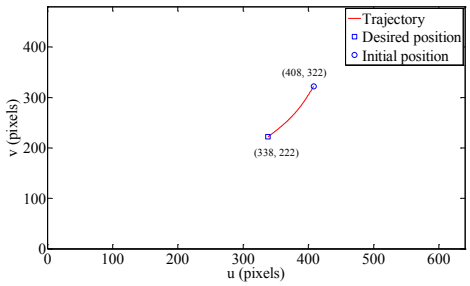
Camera 1



Camera 4



Camera 2



Camera 3

Figure 4-10 Trajectories in the image planes

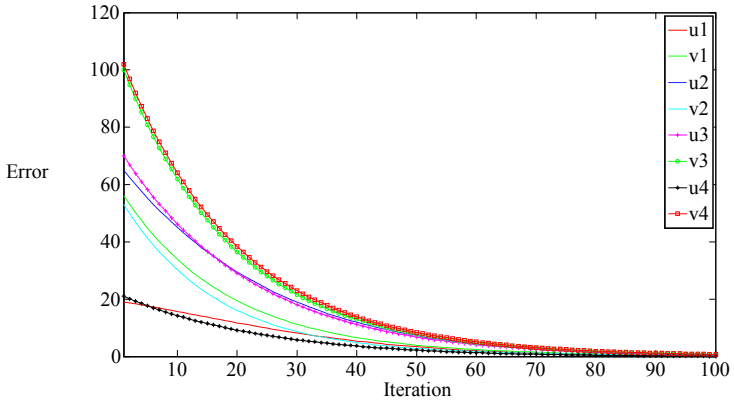


Figure 4-11 Image feature errors in the image plane

Figure 4-9 shows that the workpiece approaches the desired position from the initial position 1 in the 3D robot base frame and Figure 4-10 shows the trajectories of the image features in the image frame. Figure 4-11 shows that the errors of the image feature converge to zero asymptotically.

Case 2

The initial position of the robot end effector frame in the robot base frame is represented in the transformational matrix as follows

$$ {}^{B}_{E}T = \begin{bmatrix} 0.9781 & -0.2032 & 0.0448 & -0.0762 \\ 0.2074 & 0.9691 & -0.1338 & 0.4051 \\ -0.0162 & 0.1401 & 0.9900 & 0.3179 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The control objective is to drive the workpiece from the initial position to the desired position. The simulation results are shown in Figure 4-12 to Figure 4-14.

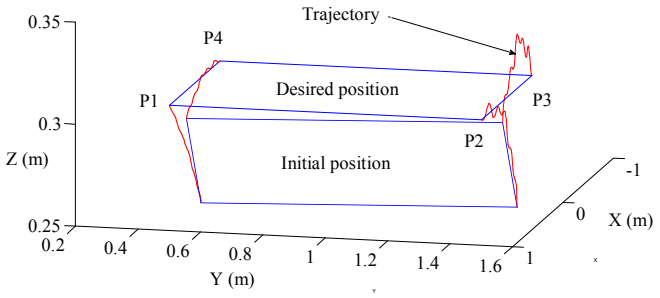


Figure 4-12 Trajectories of points P1, P2, P3 and P4 in the robot base frame

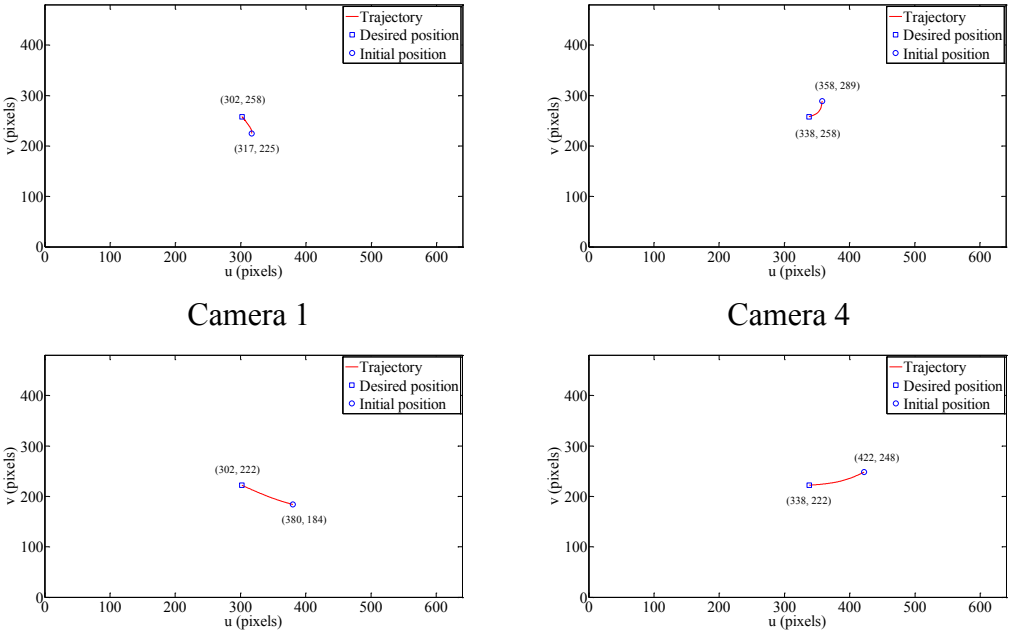


Figure 4-13 Trajectories in the image planes

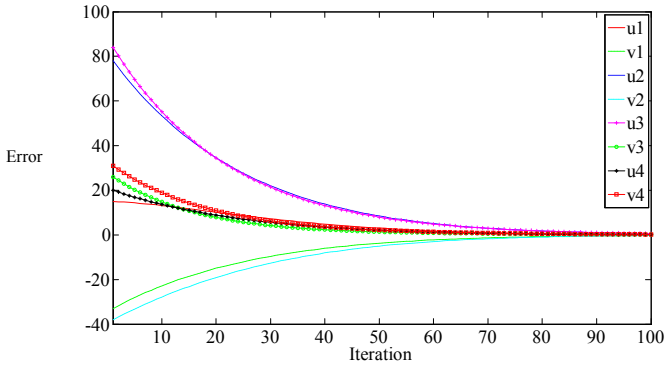


Figure 4-14 Image feature errors in the image plane

Figure 4-12 shows that the workpiece approaches the desired position from the initial position 2 in the 3D robot base frame, and Figure 4-13 shows the trajectories of the image features in the image frame. Figure 4-14 shows that the errors of the image feature converge to zero asymptotically.

4.1.3 Error analysis

From the simulation results of case 1 and 2, the maximum error of the image features in the image frames of four cameras at the end of the visual servoing process is calculated as follows

$$\Delta e = \sqrt{\Delta u_{\max}{}^2 + \Delta v_{\max}{}^2} \text{ (pixels)}$$

Hence, in case 1, the maximal error of image features $\Delta e_1 = 1.2$ (pixels). Similarly, in case 2, $\Delta e_2 = 0.9$ (pixels). If the positioning error of the workpiece with respect to the corresponding target point in the robot base frame is denoted as Δx, the relationship between Δx and Δe is proximately described as

$$\lambda \frac{\Delta x}{Z_0} \alpha_x = \Delta e (\Delta e = \max\{\Delta e_1, \Delta e_2\})$$

.

Thus, the positioning error of the workpiece in the robot base frame is

$$\Delta x = \frac{\Delta e Z_0}{\lambda \alpha_x} = 0.89 \text{ (mm)}$$

where, λ, s_x, Z_0 are defined in Section 4.1- Introduction.

The positioning error of the reference points on the workpiece in the robot base frame is within 1 (mm), which can satisfy the need for high precision manufacturing and manipulating. The simulation results also demonstrate that the proposed control algorithm can drive the robot or the workpiece from different initial positions to the desired position. The errors of the image features converge to zero asymptotically in visual servoing.

From the simulation results, it is clear that the positioning error of the workpiece in the robot base frame of this configuration is 0.89 (mm), which is larger than 0.15 (mm), the positioning error of the multiple camera visual servoing system with a 3D CAD laser projector proposed in Chapter 3. The iteration number of visual servoing before reaching the desired position is 100, which is larger than 40 the iteration number of the proposed configuration in Chapter 3. It is clear that the multiple camera configuration in Chapter 3 is better than the configuration proposed in this chapter in terms of positioning precision and convergence speed of image errors in the image frames. On the other hand, compared with the multiple camera configuration in Chapter 3, it has some advantages such as its simple structure, cost-effectiveness, and easy calibration. Therefore, it is concluded that the multiple camera-multiple target point configuration is suitable for application where the robotic manufacturing system has only one working position with the demand of less precise positioning. While the multiple cameras configuration in chapter 3 is suitable for the application of robotic manufacturing system which has multiple working positions

with the demand of high precise positioning.

Conclusions

In this chapter, the multiple camera-multiple target point visual servoing system is presented. The depths of the target points are computed online by using the stereo vision method. The visual servoing scheme is successfully designed to generate control signals for the robot dynamic controller. Based on the dynamic model of the robotic manufacturing system, the robot dynamic controller is designed to drive the workpiece to achieve the desired position. The simulation results verify the effectiveness of the proposed scheme, and also validate the feasibility of applying the multiple camera-multiple target point configuration to large-scale robotic manufacturing systems in the aerospace industry.

CHAPTER 5 IMAGE-BASED VISUAL SERVOING USING IMPROVED IMAGE MOMENTS IN 6-DOF ROBOT SYSTEMS

Introduction

In Chapters 3 and 4, two configurations of multiple camera visual servoing for large-scale robotic manufacturing positioning systems were presented, in which the geometrical feature points are adopted as target features, i.e., image points in the image plane are used as image features, and multiple camera configurations address the challenge of depth determination to improve the control performance of visual servoing systems.

In this chapter, image moments are used as image features to decouple the components of the camera velocity screw and to reduce the singularities of the interaction matrix and the local minimas of image features in visual servoing, and thus to improve the control performance of the robot tracking system. Meanwhile IBVS is still adopted as visual servoing strategy.

Compared with other visual servoing methods, IBVS has three main advantages [24]. Firstly, IBVS is a "model-free" method, which means that it does not require the model of the target object. Secondly, IBVS is robust to camera model errors. Lastly IBVS is also insensitive to camera calibration errors [82]. However, the drawbacks of IBVS are the singularities of interaction matrix and the local minimas of the image features, which lead to IBVS failure. Thereby, the choice of image features is a key point to solve the problem of the singularity of interaction matrix and the local minima of the image

features. Several efforts have been made to determine some decoupling image features to deliver a triangular or diagonal interaction matrix [21, 22, 28].

The image moments are normally used for pattern-recognition in computer vision [31, 32]. Using proper image moments as image features in visual servoing renders the interaction matrix with a maximal decoupled structure [22]. Thus, the inherent problem of singularities of the interaction matrix is avoided and the control performance of IBVS system is significantly improved. In [22], based on image moment invariants to 2-D translation, 2-D rotation, and scale, two image moments S_x and S_y are selected as image features to control ω_x and ω_y the angular velocities around the *x* and *y* axes of the camera frame respectively for the central symmetrical target object. However, the simulation results show that these two image features S_x and S_y cannot represent the pose of the target object all the time.

In this chapter, two new improved image features are proposed as image features for central symmetrical target objects to control their pose rotating around the *x* and *y* axes of the camera frame respectively. A dynamic visual servoing controller is designed based on the proposed image features to efficiently drive the robot end-effector to track the desired object. The developed controller is applied to a 6-DOF robot visual servoing tracking system and the experimental results demonstrate the effectiveness of the proposed scheme.

IBVS using image moments

The objective of IBVS is to control the robot end-effector to approach an unknown target object with various pose and shape. In this part, the development of the IBVS using

image moments for a 6-DOF robot visual servoing tracking system is introduced. The configuration of the considered system is shown in Figure 5-1, which is composed of a 6-DOF robot and a camera mounted on the robot end-effector.

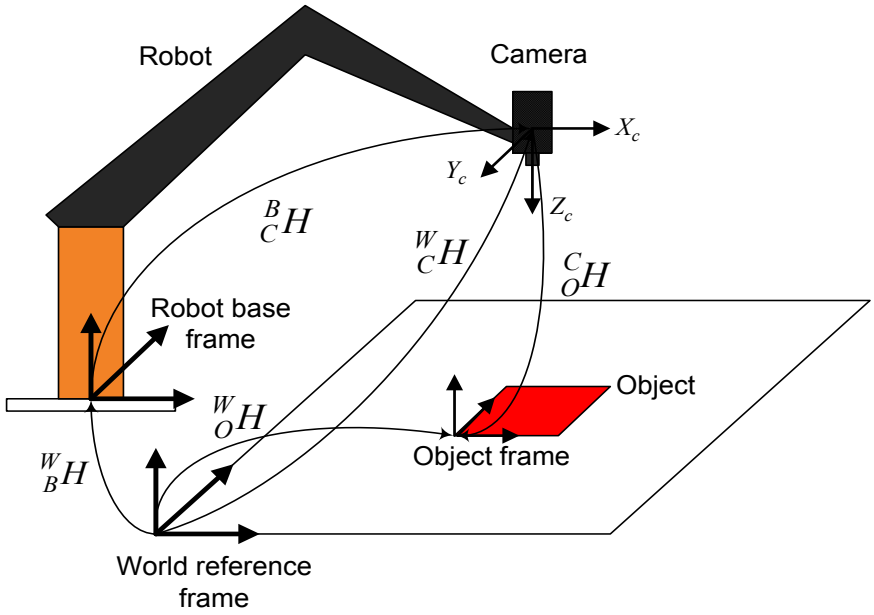


Figure 5-1 Robotic eye-in-hand system configuration

In Figure 5-1, H denotes the transformation between two reference frames. To accomplish IBVS for such a robot visual servoing tracking system, we firstly derive the interaction matrix, which indicates the relationship between the motion of selected image features and the screw velocity of the camera based on the six chosen image features and then design an IBVS controller to control the motion of the robot.

5.1.1 Image feature extraction

In order to control the 6 DOF of the camera, at least six image features are needed for visual servoing. For an image $I(x,y)$, the two-dimensional geometric moment m_{ij} and central moments μ_{ij} of order $i+j$ are defined as [22]

$$m_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^i y^j I(x,y)dxdy \tag{5.1}$$

$$\mu_{ij} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-x_g)^i (y-y_g)^j I(x,y)dxdy \tag{5.2}$$

where (x_g, y_g) are the coordinates of the centroid in the image frame.

It is known that the low-order moments have their own properties, which can denote the geometric characteristics of the target object in the image. Four image features are chosen as the same as those in [22], i.e.

$a = m_{00}$: the area of the object in the image plane (zeroth order moment);

$x_g = m_{10}/m_{00}$, $y_g = m_{01}/m_{00}$: the coordinate of the centroid (first order moments);

$\phi = \frac{1}{2}\arctan(\frac{2\mu_{11}}{\mu_{20}-\mu_{02}})$: the orientation angle (second order moments) shown in Figure 5-2.

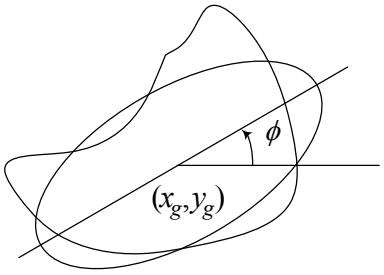


Figure 5-2 Orientation of an object defined as the orientation of the ellipse obtained using the value of the object moments of order less than 3

However, there are many ways to choose the image features. In [83], the well-known skewnesses are chosen as the rest two image features. In [22], the image moments $\begin{cases} S_x = (c_2c_3 + s_2s_3)/K \\ S_y = (s_2c_3 - c_2s_3)/K \end{cases}$ have been utilized as image features for the central symmetrical target object to control ω_x and ω_y, the angular velocities around the x and y axes of the camera frame respectively, which obtains the decoupled visual servoing behavior. Nevertheless, our simulation results show that S_x and S_y cannot represent the right pose of the target object all the time.

In this chapter, two new image moments are proposed as image features to replace S_x and S_y in [22] if target objects have central symmetrical shapes, which can tell the right poses of the target objects. The final improved image features are defined as follows

$$\begin{cases} P_x = 0.1 - (c_1c_2 + s_1s_2)/M_1^{\frac{9}{4}} \\ P_y = (s_1c_2 - c_1s_2)/M_1^{\frac{9}{4}} \end{cases} \tag{5.3}$$

where $c_1 = \mu_{20} - \mu_{02}$, $c_2 = \mu_{03} - 3\mu_{21}$, $s_1 = \mu_{11}$, $s_2 = \mu_{30} - 3\mu_{12}$, $M_1 = \mu_{20} + \mu_{02}$.

Table 5-1 The error of image features P_x (e_{P_x})

e_{P_x}		β						
		-15°	-10°	-5°	0°	$+5^\circ$	$+10^\circ$	$+15^\circ$
α	-15°	–0.0078	–0.0061	–0.0015	–0.0026	–0.0042	–0.0061	–0.0062
	-12°	–0.0048	–0.0041	–0.0019	–0.0019	–0.0037	–0.0051	–0.0050
	-9°	–0.0040	–0.0029	–0.0030	–0.0023	–0.0040	–0.0029	–0.0030
	-6°	–0.0026	–0.0020	–0.0016	–0.0011	–0.0016	–0.0020	–0.0024
	-3°	–0.0012	–0.0014	–0.0015	–0.00076	–0.00083	–0.0011	–0.00097
	0°	–0.000087	–0.000051	–0.00001	0	0.000051	0.000075	0.000091
	$+3^\circ$	0.0012	0.0014	0.0015	0.00076	0.00079	0.0010	0.00099
	$+6^\circ$	0.0025	0.0022	0.0016	0.0011	0.0016	0.0022	0.0023
	$+9^\circ$	0.0039	0.0029	0.0028	0.0023	0.0039	0.0029	0.0030
	$+12^\circ$	0.0044	0.0041	0.0020	0.0018	0.0037	0.0041	0.0049
	$+15^\circ$	0.0062	0.0061	0.0013	0.0026	0.0041	0.0061	0.0062

Table 5-2 The error of image features P_y (e_{P_y})

e_{P_y}		α						
		-15°	-10°	-5°	0°	$+5^\circ$	$+10^\circ$	$+15^\circ$
β	-15°	−0.0105	−0.0059	−0.0072	−0.0066	−0.0069	−0.0058	−0.010
	-12°	−0.0074	−0.0051	−0.0109	−0.0051	−0.0046	−0.0051	−0.0074
	-9°	−0.0055	−0.0056	−0.0053	−0.0044	−0.0033	−0.0051	−0.0074
	-6°	−0.0035	−0.0032	−0.0037	−0.0039	−0.0025	−0.0023	−0.0029
	-3°	−0.0012	−0.0013	−0.0019	−0.0019	−0.0013	−0.0012	−0.0014
	0°	−0.000028	−0.000012	−0.000145	0	0.00013	0.00001	0.000056
	$+3^\circ$	0.0012	0.0013	0.0019	0.0019	0.0014	0.0012	0.0014
	$+6^\circ$	0.0037	0.0031	0.0037	0.0037	0.0025	0.0023	0.0030
	$+9^\circ$	0.0055	0.0052	0.0054	0.0044	0.0034	0.0029	0.0067
	$+12^\circ$	0.0075	0.0053	0.0108	0.0054	0.0050	0.0047	0.0079
	$+15^\circ$	0.0105	0.0058	0.0075	0.0066	0.0069	0.0058	0.0098

Table 5-3 The error of image features S_x (e_{S_x})

e_{S_x}		β						
		-15°	-10°	-5°	0°	$+5^\circ$	$+10^\circ$	$+15^\circ$
α	-15°	0.0346	0.0047	0.0092	0.0162	0.0197	0.0248	0.0341
	-12°	0.0344	0.0341	0.0077	0.0171	0.0169	0.0250	0.0338
	-9°	0.0189	0.0195	0.0133	0.0134	0.0177	0.0195	0.0191
	-6°	0.0189	0.0134	0.0092	0.0096	0.0092	0.0135	0.0192
	-3°	0.0128	0.0089	0.0089	0.0067	0.0050	0.0660	0.0144
	0°	−0.00046	−0.0002	−0.000032	0	0.00005	0.00038	0.000046
	$+3^\circ$	−0.0127	−0.0092	−0.0085	−0.0067	−0.0046	−0.0070	−0.0144
	$+6^\circ$	−0.0183	−0.0034	−0.0095	−0.0096	−0.0095	−0.0144	−0.0189
	$+9^\circ$	−0.0178	−0.0195	−0.0126	−0.0134	−0.0179	−0.0195	−0.0192
	$+12^\circ$	−0.0318	−0.035	−0.0077	−0.017	−0.0070	−0.0251	−0.0305
	$+15^\circ$	−0.0362	−0.045	−0.090	−0.0162	−0.0192	−0.0249	−0.0339

Table 5-4 The error of image features S_y (e_{S_y})

e_{S_y}		α						
		-15°	-10°	-5°	0°	$+5^\circ$	$+10^\circ$	$+15^\circ$
β	-15°	0.0285	0.0231	0.0442	0.0331	0.0310	0.023	0.0287
	-12°	0.0206	0.0233	0.0598	0.0545	0.0478	0.0216	0.0199
	-9°	0.0322	0.0211	0.0269	0.0177	0.0168	0.019	0.0307
	-6°	0.0121	0.0155	0.0191	0.0155	0.0136	0.0107	0.00944
	-3°	0.0033	0.00994	0.0099	0.0080	0.0071	0.005	0.0025
	0°	−0.00013	−0.000022	0.0012	0	0.0012	0.000022	0.00037
	$+3^\circ$	−0.003	−0.0102	0.0091	0.0087	0.0104	−0.005	−0.0025
	$+6^\circ$	0.011	0.00056	−0.0057	0.0012	−0.001	0.0048	0.0131
	$+9^\circ$	0.0029	−0.0198	−0.00276	0.0048	−0.0175	−0.0179	0.0015
	$+12^\circ$	−0.0209	−0.0230	−0.00387	−0.0269	−0.0272	−0.024	−0.0199
	$+15^\circ$	−0.0251	−0.0278	−0.00446	−0.0371	−0.0305	−0.0228	−0.0233

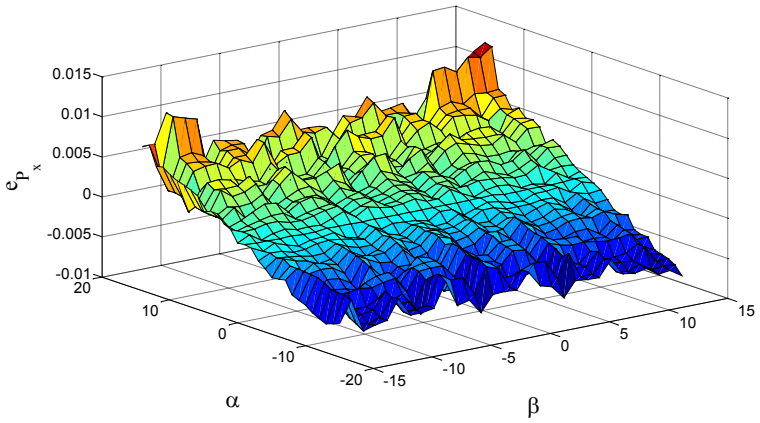


Figure 5-3 Representation of errors e_{P_x} of the image feature P_x on $[-15°;+15°] \times [-15°;+15°]$ for a rectangular object

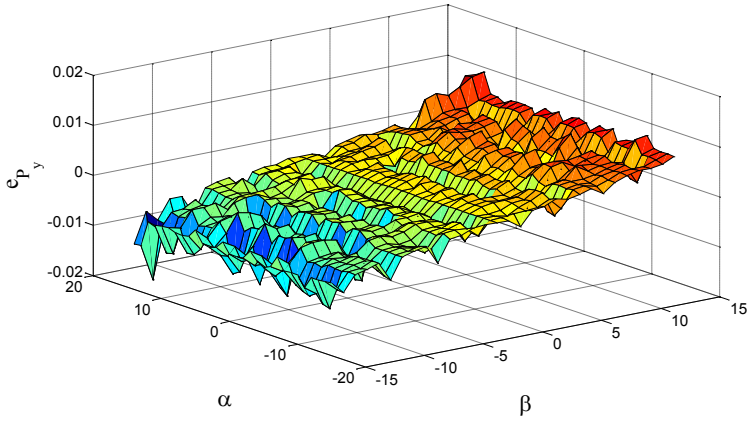


Figure 5-4 Representation of errors e_{P_y} of image feature P_y on $[-15°;+15°] \times [-15°;+15°]$ for a rectangular object

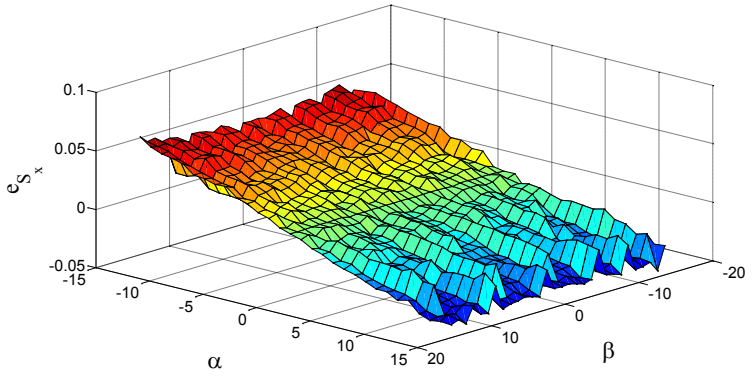


Figure 5-5 Representation of errors e_{S_x} of image feature S_x on $[-15°;+15°] \times [-15°;+15°]$ for a rectangular object

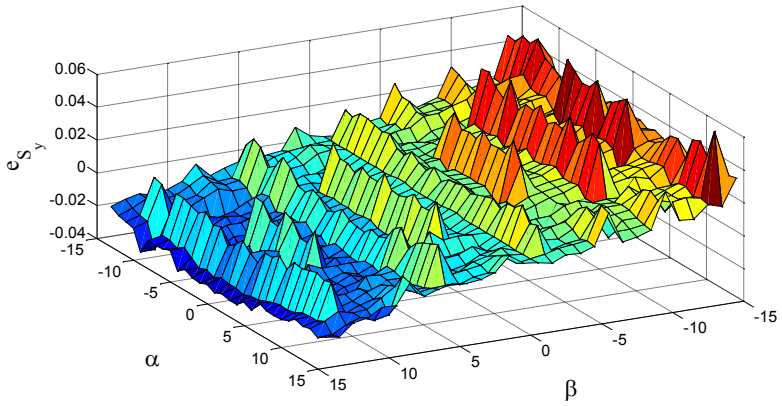


Figure 5-6 Representation of errors e_{S_y} of image feature S_y on $[-15°;+15°] \times [-15°;+15°]$ for a rectangular object

To validate the proposed image features, the simulations had been carried out to test that the values of the proposed image features could roughly change linearly with the pose of the target object. For example, assume that the error of the proposed image feature relating to the pose of the rotation around the x axis by $+10°$ from the initial position is A. Then if this object has the pose with the rotation around the x axis by $-10°$ from the initial position, the error of the proposed image feature should be $-A$. Table 5-1 to Table 5-4 show the relationships between the errors of the image features and the rotational angles of the target object from the initial position around the x and y axes of the camera frame, where α and β represent the rotational angles around the x and y axes respectively. Figure 5-3 to Figure 5-6 are the representations of the errors of the image features with respect to α and β in Table 5-1 to Table 5-4 respectively. Figure 5-3 to Figure 5-4 show e_{P_x}, e_{P_y} the errors of the proposed image features in (5.3) with different poses for a normal rectangular object. Figure 5-5 to Figure 5-6 show e_{S_x}, e_{S_y} the errors of two image features denoted by S_x, S_y defined in [22].

Referring to Table 5-1 and Table 5-2, it is clear that if β is a fixed value, e_{P_x} only changes with respect to α， and if α is a fixed value, e_{P_y} only changes with respect to β. Hence, one can conclude that P_x is almost independent of ω_y, and P_y is almost independent of ω_x. For instance, if β is 0, e_{P_y} is 0 and if β is $+9°$, e_{P_y} stays around 0.0041 no matter what value α takes from $-15°$ to $+15°$.

Table 5-3 and Table 5-4 show that S_x and S_y are almost independent of ω_x and ω_y respectively. However, the feature S_y cannot represent the correct pose. In general, for the opposite pose, the control signals should have an opposite sign to drive the robot to the desired position, which demands the feedback errors to have different signs. From Table 5-4, it is clear that S_y cannot satisfy this condition, i.e., S_x and S_y cannot produce the effective controller. The analysis above is based on the rectangular-shaped object but the conclusion of the analysis is applicable for other central symmetrical-shaped objects, such as star, date core, etc. So the proposed image features in equation (5.3) can represent the right pose of the object and hence the visual servoing based on these image features is expected to produce the proper control signal for the robot.

In summary, six image features are chosen as: $s = \begin{bmatrix} a & x_g & y_g & \varphi & P_x & P_y \end{bmatrix}^T$. The interaction matrices corresponding to a, x_g, y_g, and ϕ are rewritten here (the details are referred to in [22])

$$J_a = \begin{bmatrix} 0 & 0 & \frac{2}{Z}m_{00} & \frac{3}{\lambda}m_{01} & -\frac{3}{\lambda}m_{10} & 0 \end{bmatrix} \tag{5.4}$$

$$\begin{cases} J_{xg} = \begin{bmatrix} -\dfrac{\lambda}{z} & 0 & \dfrac{x_g}{z} & \dfrac{4m_{11}}{\lambda m_{00}} - \dfrac{3x_g y_g}{\lambda} & -\dfrac{4m_{20}}{\lambda m_{00}} - \lambda + \dfrac{3x_g{}^2}{\lambda} & y_g \end{bmatrix} \\ J_{yg} = \begin{bmatrix} 0 & -\dfrac{\lambda}{z} & \dfrac{y_g}{z} & \dfrac{4m_{02}}{\lambda m_{00}} + \lambda - \dfrac{3y_g{}^2}{\lambda} & -\dfrac{4m_{11}}{\lambda m_{00}} + \dfrac{3x_g y_g}{\lambda} & -x_g \end{bmatrix} \end{cases} \tag{5.5}$$

$$J_{\phi} = \frac{1}{\Delta^2 + 4\mu_{11}{}^2}(\Delta J_{\mu 11} - \mu_{11} J_{\Delta}) \tag{5.6}$$

The interaction matrices corresponding to P_x, P_y are derived as follows:

$$\begin{cases} J_{Px} = -(c_2 J_{c1} + c_1 J_{c2} + s_2 J_{s1} + s_1 J_{s2}) / M_1^{\frac{9}{4}} + \dfrac{9}{4}(c_1 c_2 + s_1 s_2) J_{M1} / M_1^{\frac{13}{4}} \\ J_{Py} = (c_2 J_{s1} + s_1 J_{c2} - s_2 J_{c1} - c_1 J_{s2}) / M_1^{\frac{9}{4}} - \dfrac{9}{4}(s_1 c_2 + c_1 s_2) J_{M1} / M_1^{\frac{13}{4}} \end{cases} \tag{5.7}$$

where $J_{\Delta} = J_{\mu 20} - J_{\mu 02}$, $J_{\mu 11}$, $J_{\mu 02}$, $J_{\mu 20}$, J_{c1}, J_{c2}, J_{s1}, J_{s2}, are referred to in [22]. Based on the chosen six image features, the overall interaction matrix is calculated by

$$J_{image} = \begin{bmatrix} J_a & J_{xg} & J_{yg} & J_{\phi} & J_{Px} & J_{Py} \end{bmatrix}^T \tag{5.8}$$

An IBVS control algorithm is developed to improve the tracking performance for the robot tracking system.

IBVS controller design

Based on the derived interaction matrix, an IBVS controller is designed to generate the feedback control signal for IBVS, which is shown in Figure 5-7. The controller aims at driving the robot end-effector to approach an unknown object with different poses and shapes. In the proposed algorithm, the visual servoing is accomplished in three stages as follows:

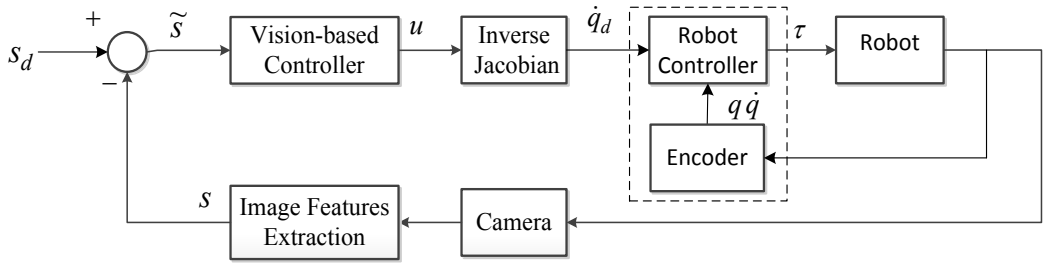


Figure 5-7 IBVS control diagram

5.1.2 Teaching stage

This stage is accomplished manually. The image of the target object is taken in the desired position of the robot end-effector and all six desired image features s_d are recorded, which will be used as the reference of $\tilde{s}$ values of the image features. For the same object, this stage needs to be performed only once.

5.1.3 Design the IBVS control law

The desired image features are denoted as $s_d = \left[a_d \; x_{gd} \; y_{gd} \; \phi_d \; s_{xd} \; s_{yd}\right]^T$ and the current values of the image features are denoted as $s = \left[a \; x_g \; y_g \; \phi \; s_x \; s_y\right]^T$. Thus the errors of image features are $\tilde{s} = s - s_d$. The general form of IBVS controller is

$$u = -KJ_{image}^{-1}\tilde{s} \tag{5.9}$$

where u is the feedback control signal, K is the proportional gain, $\tilde{s}$ are the errors of image features, and J_{image} is the overall interaction matrix.

5.1.4 Design robot dynamic controller

Computed torque control is applied as the robot dynamic control scheme shown in Figure 5-8. For the robot, the general form of an n-joint robot dynamic can be written as [78]

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \Gamma_v\dot{q} + G(q) = \tau \tag{5.10}$$

where q is a vector of the robot joint variable; $M(q)$ is a positive definite, symmetric inertia matrix; $C(q,\dot{q})$ is a vector grouping the Coriolis and centrifugal joint toques; $\Gamma_v\dot{q}$ is a vector grouping the dissipative (friction) joint torques; $G(q)$ is a vector grouping the gravity joint torques; and τ is the command vector for the joint torques.

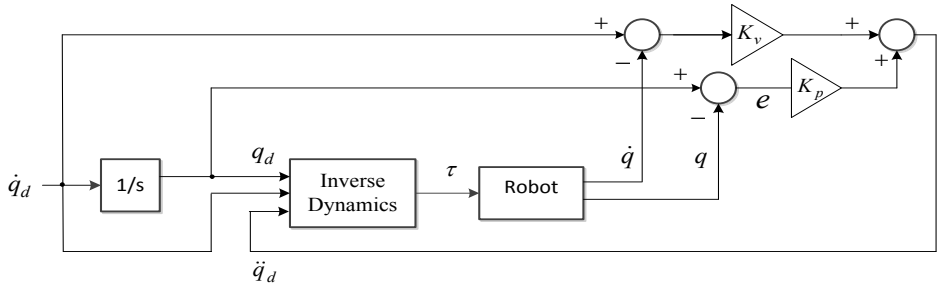


Figure 5-8 Robot dynamic controller

From the structure of the robot dynamic controller, one easily obtains the robot dynamic control law

$$\tau = M(q_d)(K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})) + C(q_d,\dot{q}_d)\dot{q}_d + \Gamma_v\dot{q}_d + G(q_d) \tag{5.11}$$

where $C(q,\dot{q})$, $\Gamma_v\dot{q}$, and $G(q)$ can be computed using the algorithm in [80]. The visual servoing control algorithm can be summarized in Figure 5-9.

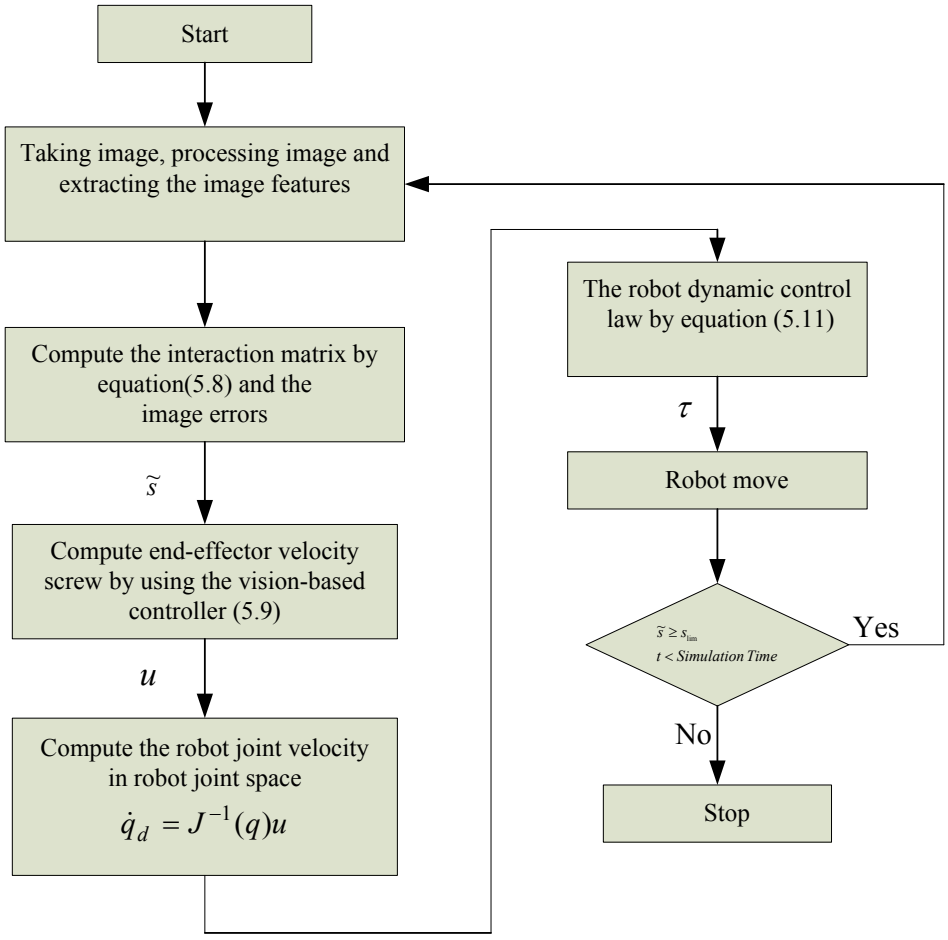


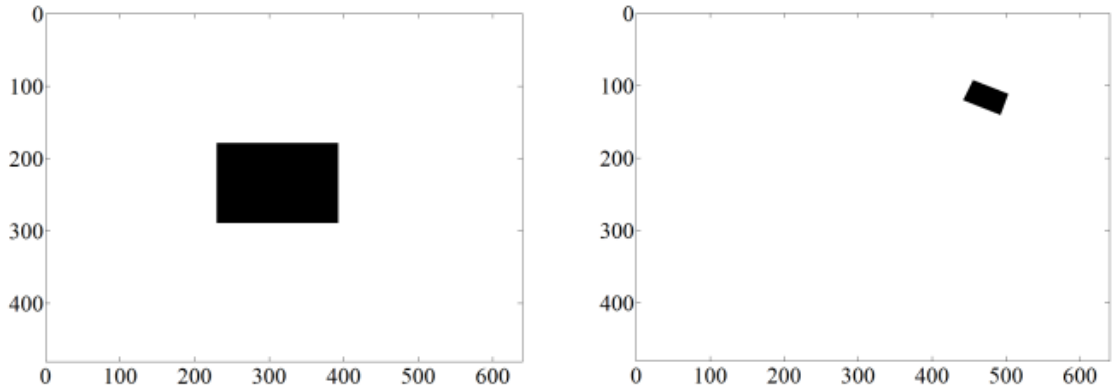
Figure 5-9 Visual servoing control algorithm diagram

Simulation studies

The proposed algorithm has been simulated in Matlab/Simulink and robotics Toolbox [80]. The robot model is adopted as PUMA 260 [84] and the camera is JAI CM-030 GE [85]. In simulation, a simple rectangular object's corner can be easily recognized in the acquired images. The desired camera position is set such that the rectangular object is parallel to the image plane at a depth of 0.15 (m). Two cases with different initial camera poses are considered.

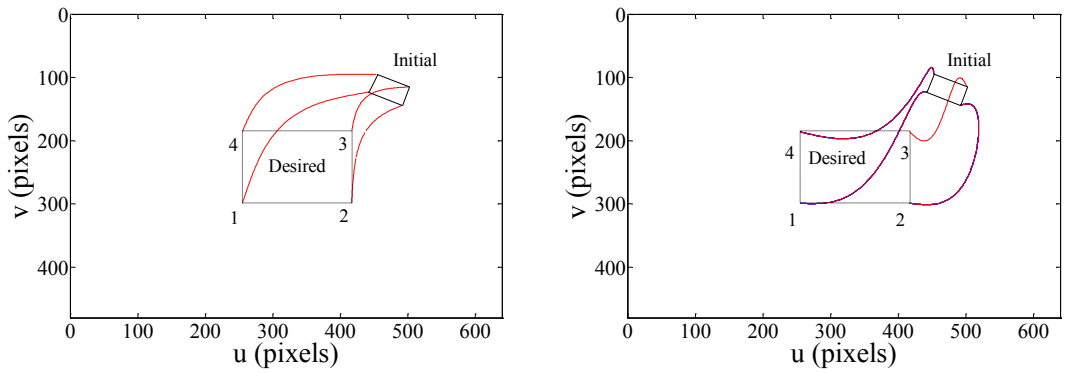
Case 1

The displacement of the camera is composed of a translation of 0.15, 0.3 and 0.4 m along the camera axes x, y, and z, respectively, and of a rotation of 10°, -20°, and -15° around these axes. The desired image and the initial image are shown in Figure 5-10. The simulation results are shown in Figure 5-11 and Figure 5-12. By comparing Figure 5-11 and Figure 5-12, one can observe that the improvements are obtained in the coordinate errors and in the trajectories of the corner points in the image frame of camera.



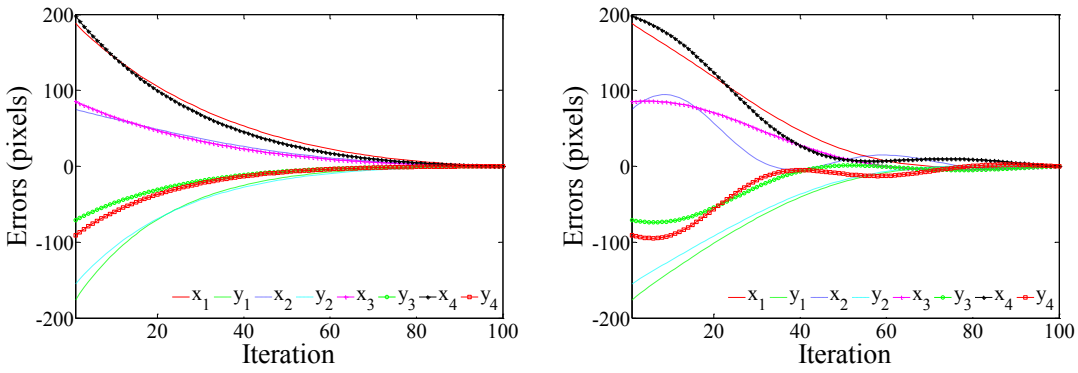
(a) (b)

Figure 5-10 Desired image (a) and initial image (b)



(a) (b)

Figure 5-11 Trajectories (a) by the proposed image features and (b) by the image features of [22]

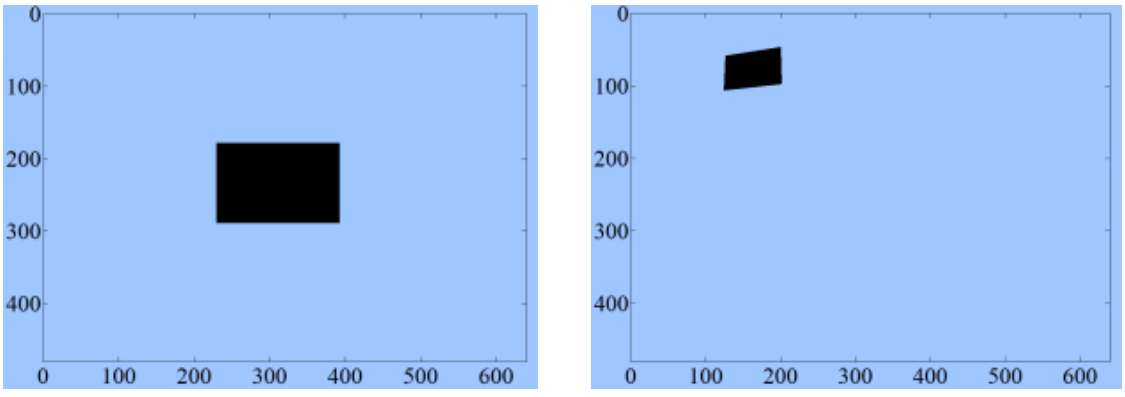


(a) (b)

Figure 5-12 Point coordinate errors (a) by the proposed image features and (b) by the image features of [22]

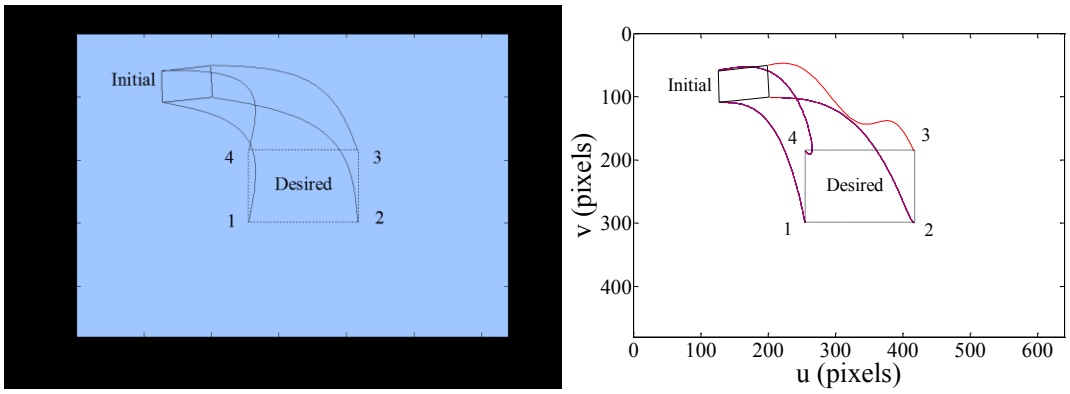
Case 2

The displacement of the camera is composed of a translation of -0.15, 0.3 and 0.15 m along the camera axes *x*, *y*, and *z*, respectively, and of a rotation of 15°, -5°, and 10° around these axes. The desired image and the initial image are shown in Figure 5-13. The simulation results are shown in Figure 5-14 and Figure 5-15. From Figure 5-14 and Figure 5-15, the improvements are obtained in the coordinate errors and the trajectories of the corner points in the image frame of the camera.



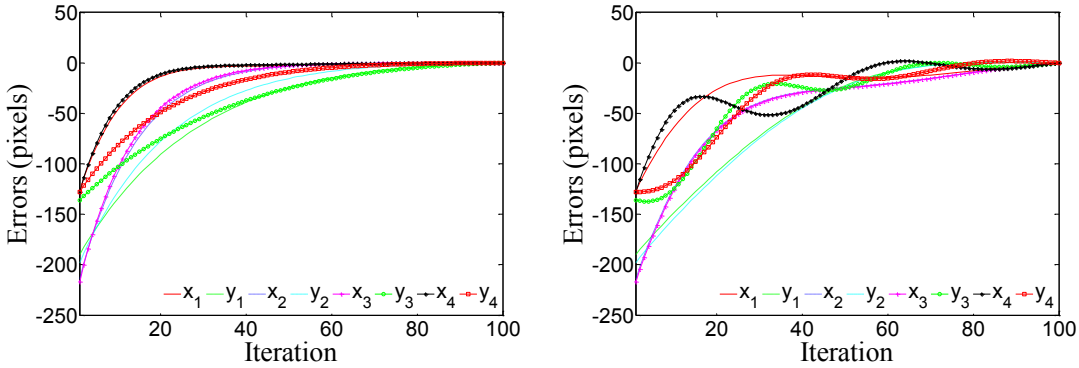
(a) (b)

Figure 5-13 Desired image (a) and initial image (b)



(a) (b)

Figure 5-14 The trajectories (a) by the proposed image features and (b) by the image features of [22]



(a) (b)

Figure 5-15 Point coordinates error (a) by the proposed image features and (b) by the image features of [22]

From simulation, it is concluded that the proposed image features in this thesis can provide better control performance than that in [22], when the rectangular target object is observed. The following experiments will show that the proposed image features are applicable for other central symmetrical-shaped objects, such as star and date core.

5.1.5 System setup

The proposed algorithm has been tested on the robot visual servoing tracking system shown in Figure 5-18, which consists of robot PUMA 200 [86], camera JAI CM-030 GE [87], and a controller unit. The system hardware components and the connections of the controller unit are shown in Figure 5-17. The camera's desired position and two initial positions in the experiments are the same as those in the simulations.

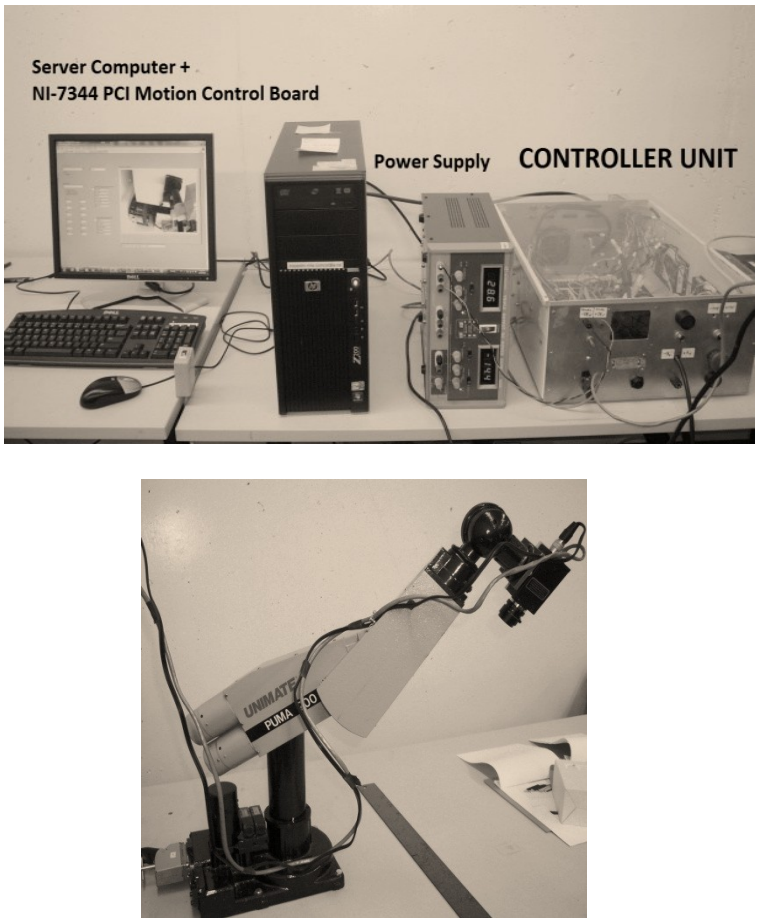


Figure 5-16 The control system and hardware connected to PUMA 260

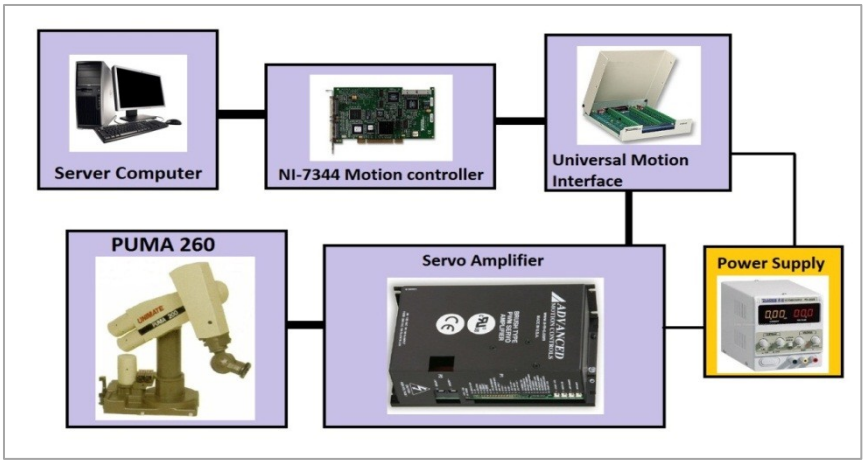


Figure 5-17 PUMA 200 hardware components and connections of controller unit

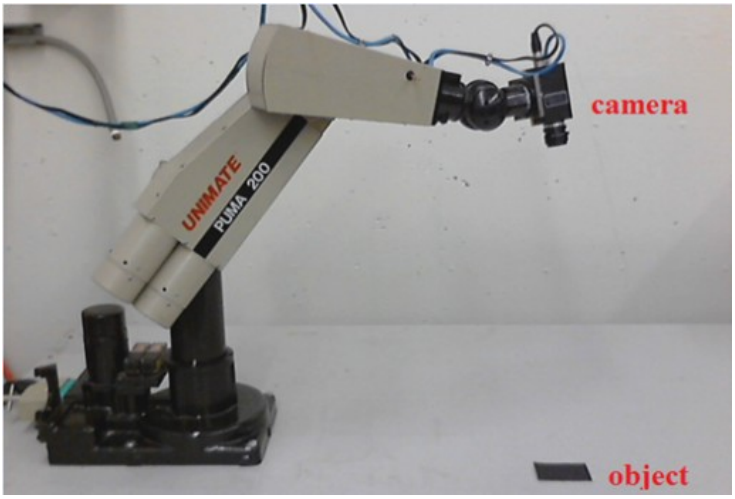


Figure 5-18 Robot visual servoing system setup

5.1.6 Experimental results

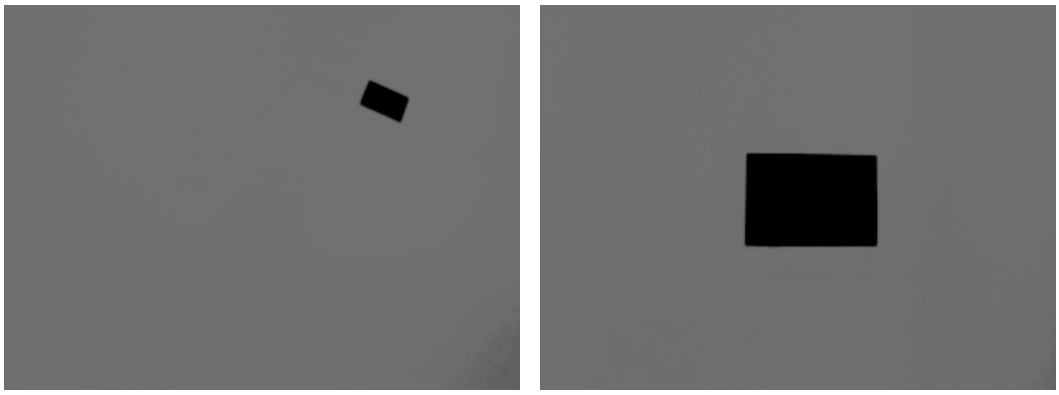
At first, the experiments will be accomplished with good camera calibration. Three different shaped target objects will be tested: rectangle, star, and date core. After that, the camera calibration error will be added into the IBVS system to test the robustness of the visual servoing system.

A. Rectangle object

Firstly, for a rectangular target object, two tasks with different initial poses of the camera are accomplished. The initial values and the desired values of the image features are provided in Table 5-5. The Figure 5-19 shows the initial and final images in task 1. The image feature errors of task 1 are displayed in Figure 5-20, in which the units of the vertical axis are different for image features. For example, for the object centroid, the unit of the vertical axis is pixel, and for the object orientation angle, the unit will be degree, and so on.

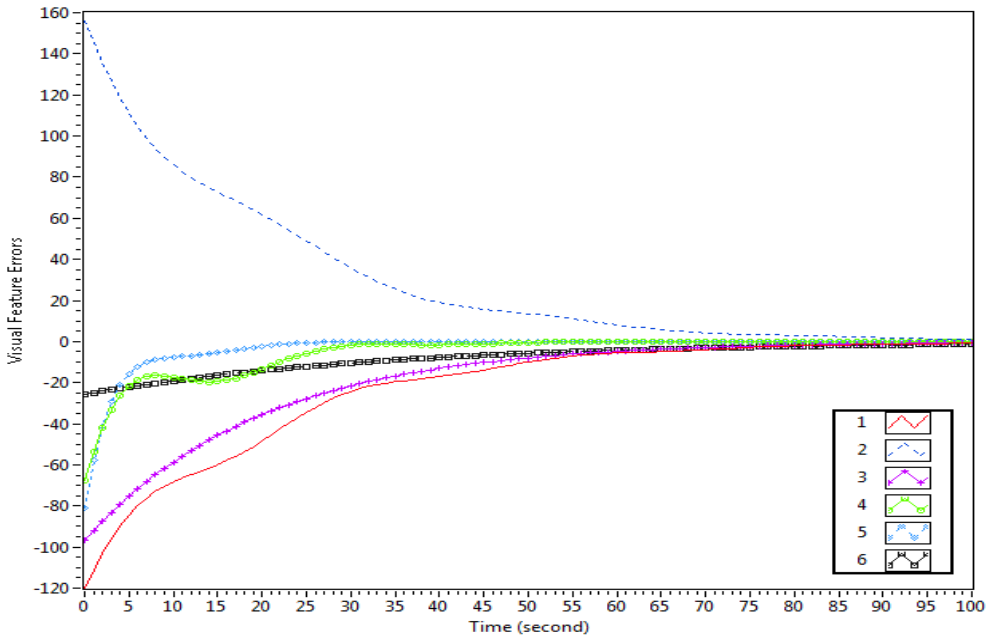
Table 5-5 The initial values and desired values of image features in task 1 and task 2

Image features	Desired values	Initial values for task 1	Errors of task 1	Initial values for task 2	Errors of task 2
x_g	0.4	-120.1	-120.5	-161.4	-161.8
y_g	-1.5	153.1	154.6	-153.5	-155
a (pixel^2)	20848	1483	-19365	3883	-16965
ϕ (degree)	90	63.9	-26.1	101	11
P_x	0.103	0.0996	-0.0034	0.096235	-0.00677
P_y	-0.0034	-0.0074	-0.0040	-0.0009	-0.0025



(a) (b)

Figure 5-19 Initial image (a) and final image (b) of task 1



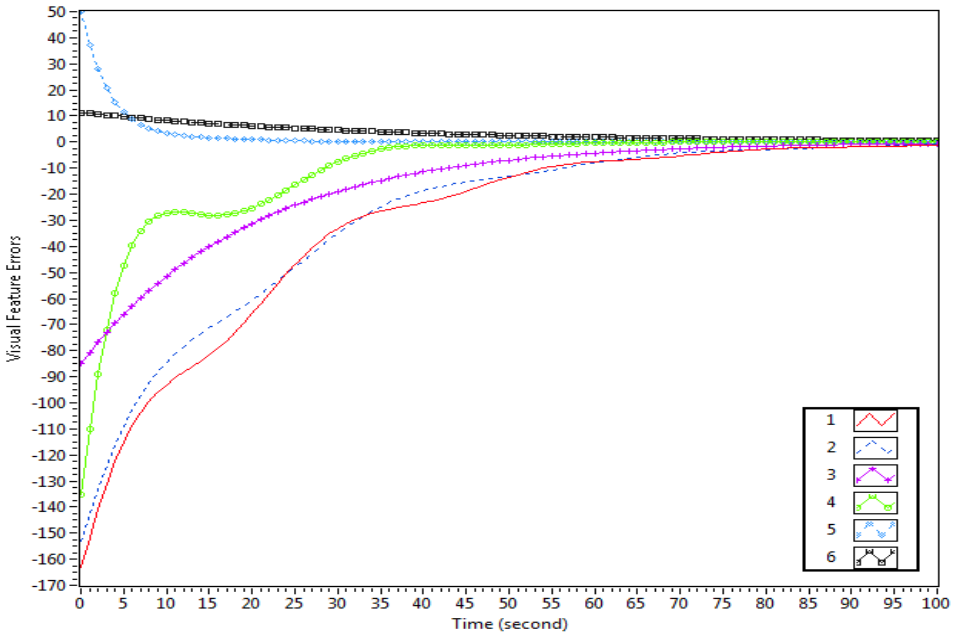
1- x_g , 2- y_g , 3- $5a\times10^{-3}$, 4- $5P_x\times10^{-5}$ 5- $5P_y\times10^{-5}$, 6- ϕ

Figure 5-20 Image feature errors of task 1

In task 2, the initial image and desired image are shown in Figure 5-21. The experimental results of task 2 are shown in Figure 5-22.

(a) (b)

Figure 5-21 Initial image (a) and final image (b) of task 2



1- x_g , 2- y_g , 3- $5a\times10^{-3}$, 4- $5P_x\times10^{-5}$ 5- $5P_y\times10^{-5}$, 6- ϕ

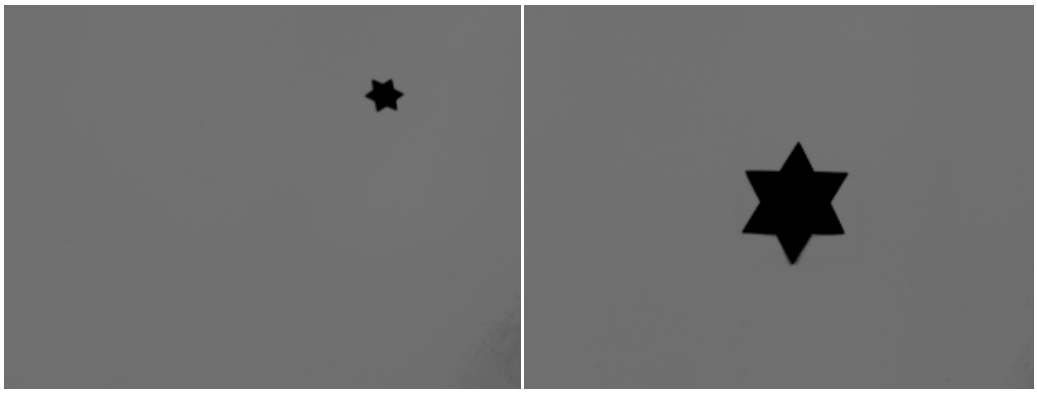
Figure 5-22 Image feature errors of task 3

B. Star and date core objects

In task 3 and task 4, the shapes of the target object are star and date core respectively. For task 3, the initial and the desired values of image features of the star-shaped object are provided in Table 5-6. The initial image and final image of task 3 are shown in Figure 5-23. The experimental results are shown in Figure 5-24.

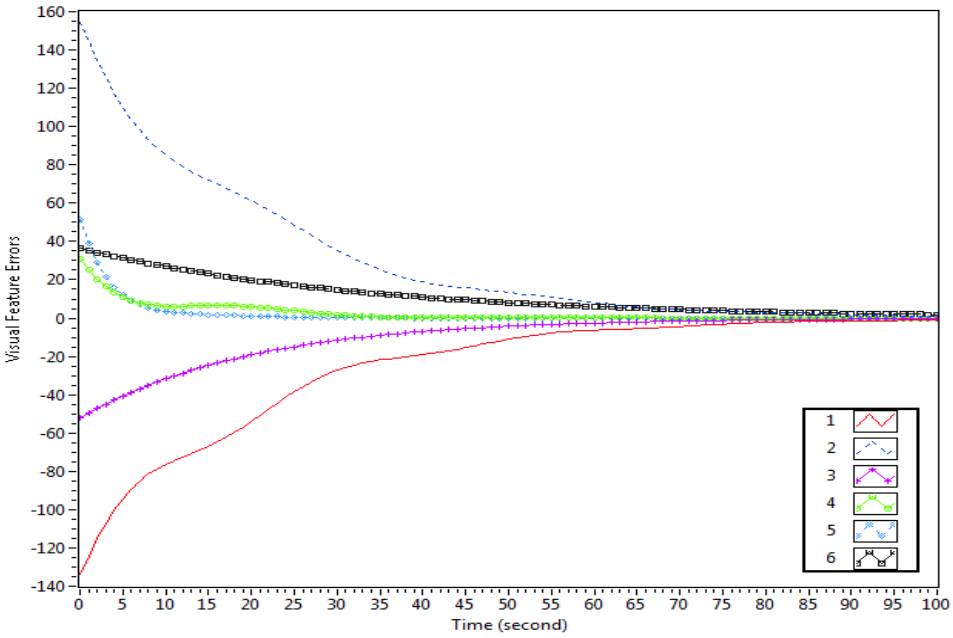
Table 5-6 The initial values and desired values of image features in task 3

Image features	Desired values	Initial values for task 3	Errors for task 3
x_g	3.5	-130.5	-134
y_g	-3.7	151	154.8
a (pixel^2)	11208	750	10458
ϕ (degree)	27	63	36
P_x	0.10019	0.10174	0.00155
P_y	-0.00002	0.00261	0.00263



(a) (b)

Figure 5-23 Initial image (a) and final image (b) of task 3



1-x_g, 2-y_g, 3-$5a\times10^{-3}$, 4-$5P_x\times10^{-5}$ 5-$5P_y\times10^{-5}$, 6-ϕ

Figure 5-24 Image feature errors of task 3

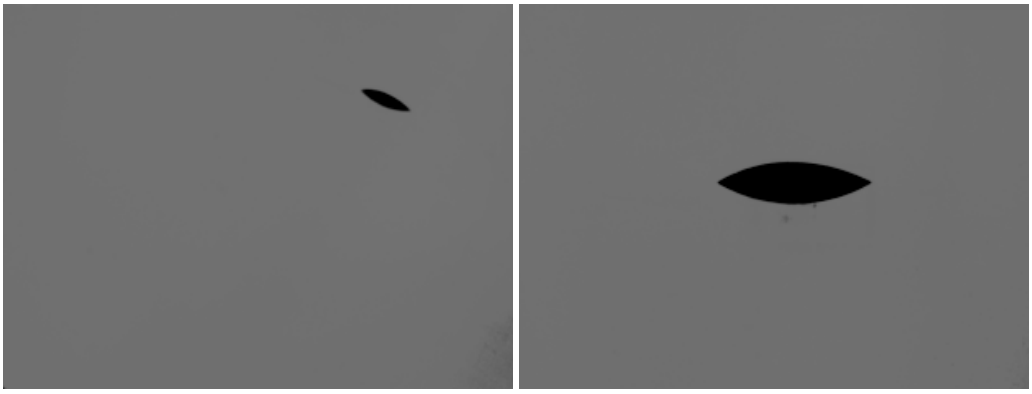
For task 4, the initial and the desired values of image features of date core are provided in Table 5-7.

Table 5-7 The initial values and desired values of image features in task 4

Image features	Desired values	Initial values for task 4	Errors for task 4

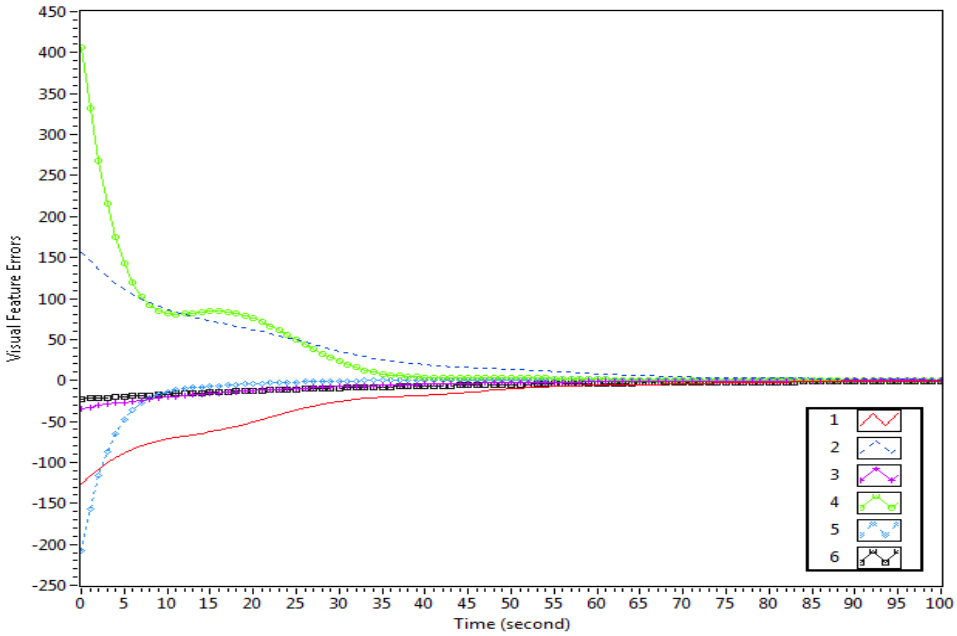
x_g	-1.1	-126.7	-127.8
y_g	-2.7	153.6	156.3
a (pixel^2)	7315	481	6834
ϕ (degree)	90	66	24
P_x	0.10815	0.12849	0.02304
P_y	0.00765	-0.02764	-0.03529

The initial image and final image of task 4 are shown in Figure 5-25. The experiment results are shown in Figure 5-26.



(a) (b)

Figure 5-25 Initial image (a) and final image (b) of task 4



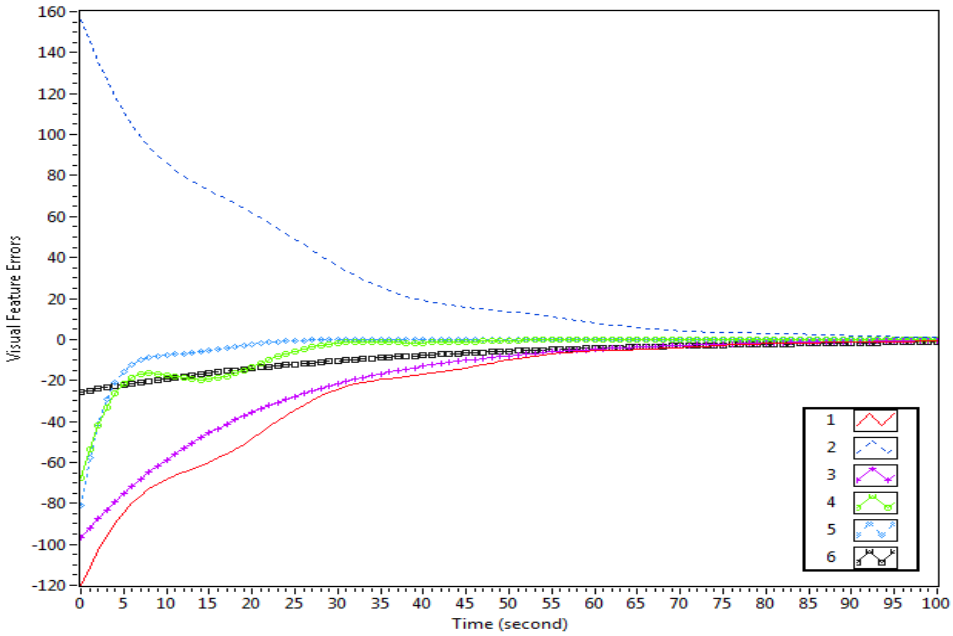
1- x_g , 2- y_g , 3- $5a\times10^{-3}$, 4- $5P_x\times10^{-5}$ 5- $5P_y\times10^{-5}$, 6- ϕ

Figure 5-26 Image feature errors of task 4

From the experimental results of task 1 through 4, it is clear that all the image feature errors approach zero, which means that the visual servoing for the tracking system is successful in all cases.

C. Robustness test

In order to test the robustness of visual servoing for the tracking system under the bad camera calibration, a 5% error focal length has been added to system. The pose and shape of the target object and initial condition are the same as those in task 1. In Figure 5-27, the image feature errors are displayed. The success of visual servoing for the tracking system demonstrates that the reasonable bad camera calibration will not affect the proposed method.



$1\text{-}x_g$, $2\text{-}y_g$, $3\text{-}5a\times10^{-3}$, $4\text{-}5P_x\times10^{-5}$ $5\text{-}5P_y\times10^{-5}$, $6\text{-}\phi$

Figure 5-27 Image feature errors of robustness test

Conclusions

In this chapter, an improved IBVS algorithm by using the image moments as the image features was proposed. Based on the chosen image features, the problem of deriving the analytical image interaction matrix describing the relationship between the motion of the camera and the velocities of the image features was addressed. In order to decouple the motion of the camera, a dynamic robot visual servoing controller was designed. A series of control signals were generated for the robot to track the target objects with various shapes and poses. The experimental results demonstrate that the tracking performance of the robotic tracking system was greatly improved.

CHAPTER 6 NEURAL NETWORK-BASED IMAGE MOMENTS FOR ROBOTIC VISUAL SERVOING

Introduction

In Chapter 5, a set of image moments were chosen for a central symmetrical target object to decouple the components of the camera velocity screw in a robotic tracking system. The control performance of the robotic tracking system has been improved because the proposed image features can correctly represent the pose of the target object in the robot base frame. However, it is noticed that the components in the interaction matrix corresponding to ω_x and ω_y (the angular velocities around the *x* and *y* axes of the camera frame respectively), are still not decoupled. In this chapter, neural networks (NN) are used to address the challenge of completely decoupling ω_x and ω_y, such that the overall interaction matrix has a diagonal form corresponding to ω_x and ω_y, which can avoid the singularity of the interaction matrix in visual servoing. Thus the performance of the control system is improved.

It is well known that NN has a very strong ability to approach generalization and has been widely applied in function approximation and data compression, prediction, nonlinearity compensation, etc. [88]. To decouple ω_x and ω_y, two virtual image moments, which are proportional to the rotational angles around the *x* and *y* axes of the camera frame respectively, are proposed as image features in this chapter. NN is used to estimate the rotational angles of a target object around the *x* and *y* axes of the camera frame in the visual servoing process. Based on the proposed image features and the other four

commonly used image moments, the interaction matrix relating the velocities of the chosen image features to the camera's motion has the maximal decoupled structure. Thus, the singularities of the interaction matrix are avoided in IBVS. In addition, the depth of the planar-shaped target object is derived to further improve the accuracy of the interaction matrix and to ensure the global stability of the visual servoing system. A dynamic visual servoing controller is designed to drive the robot end-effector to track the target object. The effectiveness of the proposed method is demonstrated in the experimental results on a 6-DOF robot visual servoing system.

IBVS using image moments

The considered visual servoing system is shown in Figure 5-1. In this part, the development of an IBVS controller for 6-DOF robotic tracking systems is introduced based on the chosen image features. Six image features are chosen for the control scheme, and the interaction matrix of the visual servoing system is derived based on the image features.

6.1.1 Interaction matrix of image moments

To demonstrate the derivation of the interaction matrix, the two-dimensional moments m_{ij}, and the central moments μ_{ij} of order $i+j$ for a dense object in the image is rewritten here

$$m_{ij} = \iint_{R(t)} x^i y^j dxdy \tag{6.1}$$

$$\mu_{ij} = \iint_{R(t)} (x - x_g)(y - y_g) dxdy \tag{6.2}$$

where $R(t)$ represents the part of the object projection in the image obtained by the camera, and (x_g, y_g) are the coordinates of the centroid in the image frame ($x_g = m_{10}/m_{00}$, $y_g = m_{01}/m_{00}$). The alternative equation of (6.1) is given by [22]

$$m_{ij} = \iint_{R(t)} f(x,y)dxdy \tag{6.3}$$

where $f(x,y) = x^i y^j$. It is noted that only $R(t)$ relates to the time t in m_{ij}. Taking the derivative of (6.3) with respect to time and using the definition of "Contour integration" and Green's Theorem [22], one obtains

$$\dot{m}_{ij} = \iint_{R(t)} \left(\frac{\partial f(x,y)}{\partial x}\dot{x} + \frac{\partial f(x,y)}{\partial y}\dot{y} + f(x,y)\left(\frac{\partial \dot{x}}{\partial x} + \frac{\partial \dot{y}}{\partial y}\right)\right)dxdy \tag{6.4}$$

Recall that the velocities of the image feature point with coordinates $\mathbf{x} = (x,y)$ in the image plane can be expressed with the physical motion of the camera as follows

$$\begin{cases} \dot{x} = -\dfrac{\lambda}{Z}v_x + \dfrac{x}{Z}v_z + \dfrac{xy}{\lambda}\omega_x - \dfrac{x^2+\lambda^2}{\lambda}\omega_y + y\omega_z \\ \dot{y} = -\dfrac{\lambda}{Z}v_y + \dfrac{y}{Z}v_z + \dfrac{y^2+\lambda^2}{\lambda}\omega_x - \dfrac{xy}{\lambda}\omega_y - x\omega_z \end{cases} \tag{6.5}$$

where $\dot{\mathbf{r}} = \begin{bmatrix} v_x & v_y & v_z & \omega_x & \omega_y & \omega_z \end{bmatrix}^T$ is the velocity screw of the camera. If the object has a planar surface, the equation of its planar surface in the camera frame *F: X-Y-Z* at an instantaneous moment *t* is given by

$$Z = k_1(t)X + k_2(t)Y + Z_0(t)$$

where $k_1(t), k_2(t)$ are coefficients, and $Z_0(t)$ is the coordinate of the intersection point of the object plane and the optical axis of the camera. In visual servoing, one has [22]

$$\frac{1}{Z} = A(\mathrm{t})x + B(t)y + C(t) \tag{6.6}$$

where $A(t) = -\frac{k_1(t)}{\lambda Z_0(t)}$, $B(t) = -\frac{k_2(t)}{\lambda Z_0(t)}$, $C(t) = -\frac{1}{Z_0(t)}$, λ the focal length of camera, and x, y the coordinates of the point in the image plane corresponding to 3-D point (X, Y, Z) in the camera frame. Substituting (6.4), (6.5) and (6.6) and using a similar method in [22], one has

$$\dot{m}_{ij} = J_{mij}\dot{\mathbf{r}}$$

where J_{mij} is the interaction matrix, i.e.,

$$J_{mij} = \begin{bmatrix} m_{vx} & m_{vy} & m_{vz} & m_{\omega x} & m_{\omega y} & m_{\omega z} \end{bmatrix} \tag{6.7}$$

where

$$\begin{cases} m_{vx} = -i\lambda(A(\mathrm{t})m_{ij} + B(t)m_{i-1,j} + C(t)m_{i-1,j+1}) - A(t)\lambda m_{ij} \\ m_{vy} = -j\lambda(A(t)m_{i+1,j-1} + B(t)m_{ij} + C(t)m_{i,j-1}) - B(t)\lambda m_{ij} \\ m_{vz} = (i+j+3)(A(t)m_{i+1,j} + B(t)m_{i,j+1} + C(t)m_{ij}) - C(t)m_{ij} \\ m_{\omega x} = (i+j+3)\frac{1}{\lambda}m_{i,j+1} + j\lambda m_{i,j-1} \\ m_{\omega y} = -(i+j+3)\frac{1}{\lambda}m_{i+1,j} - i\lambda m_{i-1,j} \\ m_{\omega z} = im_{i-1,j+1} - jm_{i+1,j-1} \end{cases}$$

Similarly, one obtains

$$\dot{\mu}_{ij} = J_{\mu ij}\dot{\mathbf{r}}$$

where $J_{\mu ij}$ is the interaction matrix and

$$J_{\mu ij} = \begin{bmatrix} \mu_{vx} & \mu_{vy} & \mu_{vz} & \mu_{\omega x} & \mu_{\omega y} & \mu_{\omega z} \end{bmatrix} \tag{6.8}$$

where

$$
\begin{cases}
\mu_{vx} = -(i+1)\lambda A(t)\mu_{ij} - iB(t)\lambda\mu_{i-1,j+1} \\
\mu_{vy} = -jA(t)\lambda\mu_{i+1,j-1} - (j+1)\lambda B(t)\mu_{ij} \\
\mu_{vz} = -A(t)\mu_{\omega y} + B(t)\mu_{\omega x} + (i+j+2)C(t)\mu_{ij} \\
\mu_{\omega x} = (i+j+3)\dfrac{1}{\lambda}\mu_{i,j+1} + i\dfrac{1}{\lambda}x_g\mu_{i-1,j+1} + (i+2j+3)\dfrac{1}{\lambda}y_g\mu_{i,j} \\
\qquad -4i\dfrac{1}{\lambda}(\dfrac{m_{11}}{m_{00}} - x_g y_g)\mu_{i-1,j} - 4j\dfrac{1}{\lambda}(\dfrac{m_{02}}{m_{00}} - y_g^2)\mu_{i,j-1} \\
\mu_{\omega y} = -\dfrac{1}{\lambda}(i+j+3)\mu_{i+1,j} - i\dfrac{1}{\lambda}y_g\mu_{i+1,j-1} + (2i+j+3)\dfrac{1}{\lambda}x_g\mu_{i,j} \\
\qquad +4i\dfrac{1}{\lambda}(\dfrac{m_{20}}{m_{00}} - x_g^2)\mu_{i-1,j} + 4j\dfrac{1}{\lambda}(\dfrac{m_{11}}{m_{00}} - x_g y_g)\mu_{i,j-1} \\
\mu_{\omega z} = i\mu_{i-1,j+1} - j\mu_{i+1,j-1}
\end{cases}
$$

In [22, 28], *A(t)*, *B(t)* and *C(t)* are treated as constant during visual servoing, which is applicable only in the vicinity of the desired position. In fact, in real time visual servoing, *A(t), B(t)* and *C(t)* are always changing with time *t*. The calculation of *A(t), B(t)* and *C(t)* will be discussed in 6.1.8.

6.1.2 Choice of image features

The four image features are selected in the same way as in [28], i.e., *a* is the area of image ($a = m_{00}$), x_g, y_g are the coordinates of the centroid of the image on the *x* and *y* axes respectively, and ϕ is the orientation angle of the image. As shown in Figure 5-2, the orientation angle ϕ is defined as $\phi = \dfrac{1}{2}\arctan(\dfrac{2\mu_{11}}{\mu_{20} - \mu_{02}})$ [21]. The details of the interaction matrix related to these four image features are referred to in [22].

Now the other two image features m_x and m_y are determined such that the interaction matrix has a maximal decoupled property and minimal nonlinearities [28]. In fact, the last

interaction matrices corresponding to the proposed image features are expected to have such form as

$$J_{mx} = \begin{bmatrix} 0 & 0 & 0 & c_x & 0 & 0 \end{bmatrix} \tag{6.9}$$

$$J_{my} = \begin{bmatrix} 0 & 0 & 0 & 0 & c_y & 0 \end{bmatrix} \tag{6.10}$$

where parameters c_x and c_y are constant. It is very difficult for (6.9) and (6.10) to be reached by combining various moment invariants [28]. To address this challenge, it is assumed that there exist two moment invariants (to 2D translation, to 2D rotation, and to scale), which are referred to as virtual moments, and denoted as m_x and m_y respectively. In addition, it is proposed that m_x, m_y are only the function of the target object pose with respect to camera frame and have such expected forms as follows

$$m_x = f_x(\beta) = c_x \beta \tag{6.11}$$

$$m_y = f_y(\gamma) = c_y \gamma \tag{6.12}$$

where β, γ are the rotational angles around the x and y axes of the camera frame with respect to the desired position. During visual servoing, m_x and m_y are very easy to compute if β, γ are available. Hence the problem now is how to estimate β and γ based on the image obtained directly from the camera.

6.1.3 Interaction matrices related to m_x, m_y

If m_x and m_y are chosen as image features, the interaction matrices can be computed easily. From (6.11) and (6.12), the derivatives of m_x and m_y with respect to time are expressed as

$$\frac{dm_x}{dt} = c_x \frac{d\beta}{dt} = c_x \omega_x \tag{6.12}$$

$$\frac{dm_y}{dt} = c_y \frac{d\gamma}{dt} = c_y \omega_y \tag{6.13}$$

Hence the interaction matrices of these two image features have the forms of equation (6.9) and (6.10). By stacking (6.9), (6.10), and the interaction matrices related to the four image features in [28], one obtains the overall interaction matrix of six image features as follows

$$J_{image} = \begin{bmatrix} -\lambda / Z_g & 0 & x_{gvz} & x_{g\omega x} & x_{g\omega y} & y_g \\ 0 & -\lambda / Z_g & y_{gvz} & y_{g\omega x} & y_{g\omega y} & -x_g \\ -a\lambda A(t) & -a\lambda B(t) & a(3/Z_g - C(t)) & 3ay_g & -3ax_g & 0 \\ 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & c_y & 0 \\ \alpha_{vx} & \alpha_{vy} & \alpha_{vz} & \alpha_{\omega x} & \alpha_{\omega y} & -1 \end{bmatrix}$$

Estimation of the rotational angle around the x and y axes of the camera frame

6.1.4 Neural network estimation

It is noticed that the following image moments are invariant to 2D translation, to 2D rotation, and to scale [28]

$$I_1 = -\mu_{20}\mu_{02} + \mu_{11}^2$$

$$I_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$I_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

$$I_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2$$

$$I_5 = -\mu_{30}^2\mu_{03}^2 + 6\mu_{30}\mu_{21}^3 - 4\mu_{30}\mu_{12}^3 - 4\mu_{21}^3\mu_{03} + 3\mu_{21}^3\mu_{12}^3)^2$$

i.e., I_1, I_2, I_3, I_4, and I_5 called image moment invariants are only the functions of β, γ, where β, γ are the rotational angles around the *x* and *y* axes of the camera frame respectively. Hence, β, γ must be only the functions of the image moment invariant I_1, I_2, I_3, I_4, and I_5 respectively. This inspires us to use the image moment invariants to estimate β, γ. In addition, to normalize the value of the image moment invariants of the objects with the same shape in the image plane, we use several combinations of the moment invariants as follows

$$c_1 = \frac{I_1}{I_2},\ c_2 = \frac{I_2}{I_3}, c_3 = \frac{I_3}{I_4},\ c_4 = \frac{I_4}{I_5} \tag{6.15}$$

where c_1, c_2, c_3 and c_4 are still image moment invariants to 2D translation, to 2D rotation, and to scale, i.e., the c_1, c_2, c_3 and c_4 are only the functions of β, γ. It is reported that c_1, c_2, c_3 and c_4 have nonlinear relationships with β, γ [28]. Therefore a NN is applied to map the nonlinear relationships between c_1, c_2, c_3, c_4 and β, γ, i.e.,

$$\beta = f_{NNX}(c_1, c_2, c_3, c_4)$$

$$\gamma = f_{NNY}(c_1, c_2, c_3, c_4)$$

To design the NN used to estimate the values of β, γ based on c_1, c_2, c_3 and c_4, the training data sets should first be acquired from the system. As shown in

Figure 6-1, the image plane of the camera at the desired position is parallel to the target object plane, i.e., $\beta = \gamma = 0^o$. In order to easily acquire the training data sets, one can fix the origin of the camera frame on the desired position and only change the rotation angles around the *x* and *y* axes of the camera frame. The images of the planar object are taken corresponding to each specific value of β, γ in a certain range referred to as the training

area. Secondly, based on the image obtained in the first step, the image moment invariant c_1, c_2, c_3, and c_4 are computed respectively in relation to β, γ. After training data sets of NN are acquired, the NN used for mapping the relationship between the image moment invariant c_1, c_2, c_3, c_4, and β, γ can be designed. In

Figure 6-1, the frames $F_0: \mathrm{X}_0 - Y_0 - Z_0$ and $F_i : \mathrm{X}_\mathrm{i} - Y_i - Z_i$ are the camera frames at the desired position (i.e., $\beta = \gamma = 0^\circ$) and general position (i.e., $\beta \neq 0^\circ$ or $\gamma \neq 0^\circ$) respectively. Z_d is the depth of the object centroid when the camera frame is at the desired position.

The NN used for mapping the function between c_1, c_2, c_3, c_4 and β, γ is shown in Figure 6-2. A multi-layer perceptron (MLP) [84] is used to map the nonlinear relationship between the input and the output. The MLP structure is composed of one input layer, one hidden layer and one output layer. The inputs of NN are the image moment invariant c_1, c_2, c_3, c_4 and the outputs of NN are $\hat{\beta}, \hat{\gamma}$ the estimation values of rotational angles around the *x* and *y* axes of the camera frame respectively.

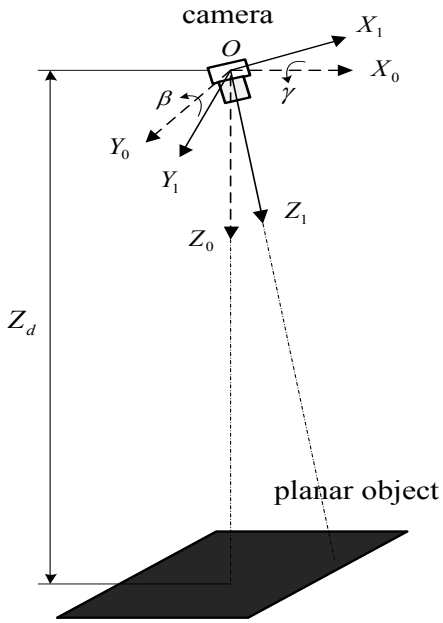


Figure 6-1 Data sets acquisition for neural network training

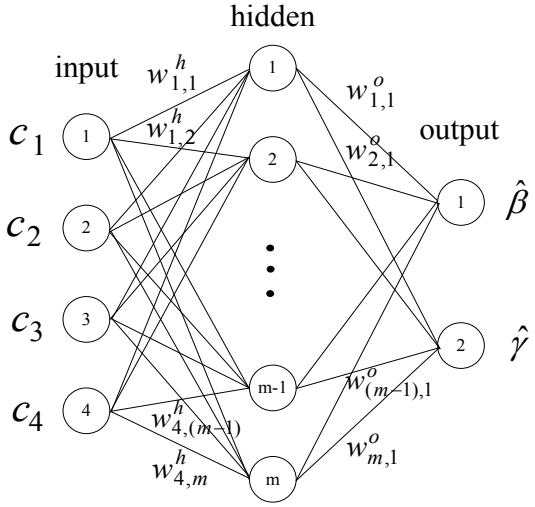


Figure 6-2 Neural network used for estimation of β and γ

The activation function used for hidden layer nodes is the hyperbolic tangent sigmoid

$$f_1(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}} \tag{6.16}$$

And the activation function used for the output layer is the linear transfer function

$$f_2(n) = n \tag{6.17}$$

where n is the net input. The nonlinear functions mapped by NN is denoted as follows:

$$\hat{\beta} = f_2(\sum_{i=1}^{m} w_{i,1}^{o} f_1(\sum_{l=1}^{m}\sum_{j=1}^{4} w_{j,l}^{h} c_j)) = \sum_{i=1}^{m} w_{i,1}^{o} f_1(\sum_{l=1}^{m}\sum_{j=1}^{4} w_{j,l}^{h} c_j) \tag{6.18}$$

$$\hat{\gamma} = f_2(\sum_{i=1}^{m} w_{i,2}^{o} f_1(\sum_{l=1}^{m}\sum_{j=1}^{4} w_{j,l}^{h} c_j)) = \sum_{i=1}^{m} w_{i,2}^{o} f_1(\sum_{l=1}^{m}\sum_{j=1}^{4} w_{j,l}^{h} c_j)$$

(6.19)

where $w_{i,1}^{o}$, $w_{i,2}^{o}$ and $w_{j,l}^{h}$ are the synopsis weights of NN.

The number of nodes for the hidden layer is fixed at m. The multiple viewpoints are generated by moving the camera with the origin of camera frame being fixed on the desired position. Based on (6.18), (6.19), to minimize the errors between the real rotation angles $\{(\beta)_i + (\gamma)_i\}$ and the output of NN, $\{(\hat{\beta})_i + (\hat{\gamma})_i\}$, the optimization objective function is defined as

$$\Delta E = \sum_{i=1}^{N_v} \{((\beta)_i - (\hat{\beta})_i)^2 + ((\gamma)_i - (\hat{\gamma})_i)^2\} \tag{6.20}$$

where N_v is the number of the training data sets. The NN training algorithm is chosen as back-propagation, based on the Levenberg-Marquarst (LM) optimization algorithm. Let $\mathbf{e}_i = \left[(\beta_i - \hat{\beta}_i) \quad (\lambda_i - \hat{\gamma}_i)\right]^T$ denote the error vector of the i^{th} input of the NN. One can convert (6.20) into the following form

$$\Delta E = \sum_{i=1}^{N_v} \mathbf{e}_i^T \mathbf{e}_i = \sum_{i=1}^{N_v} ((e_i^1)^2 + (e_i^2)^2)$$

where $\mathbf{e}_i$ is the function of the synopsis weight vector $\mathbf{w}$ of NN. Hence the ΔE is the function of $\mathbf{w}$, i.e.,

$$\Delta E(\mathbf{w}) = \sum_{i=1}^{N_v} [(e_i^1(\mathbf{w}))^2 + (e_i^2(\mathbf{w}))^2] \tag{6.21}$$

It is noted that the NN needs to be trained for different shapes of planar objects. In this thesis, the NN is trained by using Matlab 2008a Neural Networks Toolbox in experimental results.

6.1.5 NN Generalization and verification

One of the major advantages of NN is its generalization ability. The generalization of NN stands for the ability to have the outputs of the network when the inputs are not in the training set. Once the NN has been trained, it serves as an estimator of the rotational angles of the object around the *x* and *y* axes of the camera frame within the training area. To validate the effectiveness of the NN, we had carried out the experiments to validate NN in estimating the rotational angles around the *x* and *y* axes of the camera frame base on the image moment invariants. After the values of β, γ are estimated based on the NN, then virtual moments can be easily determined by (6.11) and (6.12).

6.1.6 The Depth determination

It is well known that if the image is binary, the zeroth order moment m_{00} indicates the area of the image in the image plane [21]**.** From [22], m_{00} is invariant to 2D translation and to 2D rotation, i.e., it only varies with rotational transformations around the *x* and *y*

axes of the camera frame and scale (translation along Z_0 optical axis of the camera). Hence it is known that if the angles β and γ are constant in Figure 6-3, $m_{00}(\beta,\gamma)$ will only vary with Z_i, the distance between the origin of the camera frame and the planar object along the optical axis of the camera in the desired position. The relationship between the area of the image of the target object and the depth of the target object's centroid satisfies

$$\frac{m'_{00}(\beta,\gamma)}{m_{00}(\beta,\gamma)} = (\frac{Z_i}{Z_d})^2 \quad \text{i.e., } Z_i = Z_d \sqrt[2]{\frac{m'_{00}(\beta,\gamma)}{m_{00}(\beta,\gamma)}} \tag{6.21}$$

where $m_{00}(\beta,\gamma)$ and $m'_{00}(\beta,\gamma)$ are the areas of the image of the target object corresponding to Z_d and Z_i respectively.

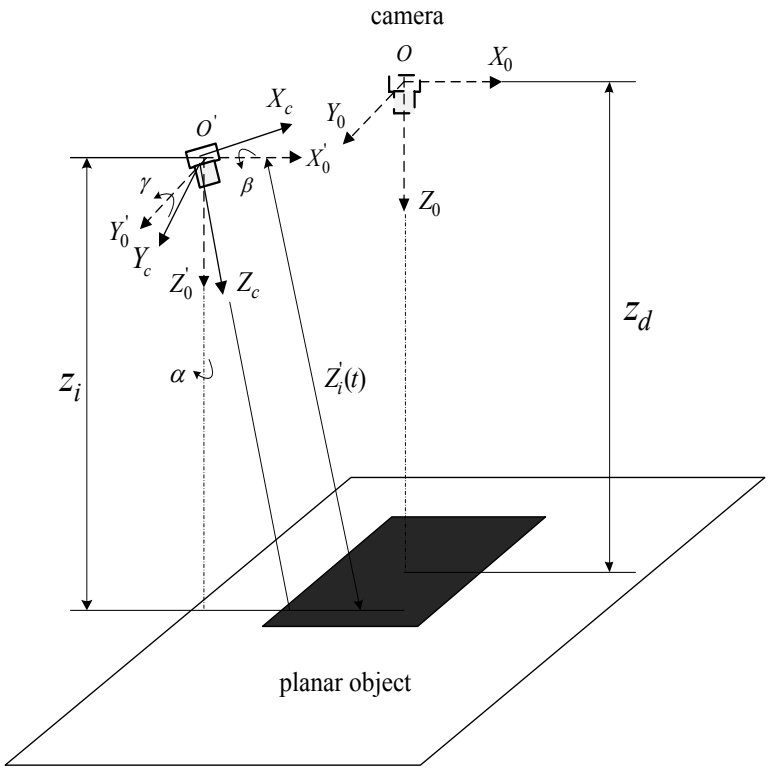


Figure 6-3 Depth estimation

6.1.7 Determination $m_{00}(\beta,\gamma)$ online

In the acquisition data sets for training the NN in 6.1.4, the image area $m_{00}(\beta,\gamma)$ corresponding to the integral values of β and γ is also collected. However the value of $m_{00}(\beta,\gamma)$ between two serial integral values of β and γ is still unknown. Hence bilinear interpolation [89] is applied to compute the image area corresponding to any value between two serial integral values of β and γ in this system. The key idea is to perform linear interpolation in one direction and then again in the other direction as shown in Figure 6-4. If the values of $m_{00}(\beta,\gamma)$ at the four integral points $Q_{11}=(\beta_1, \gamma_1)$, $Q_{12}=(\beta_1, \gamma_2)$, $Q_{21}=(\beta_2, \gamma_1)$, and $Q_{22}= (\beta_2, \gamma_2)$, the value of $m_{00}(\beta,\gamma)$ at the point $P = (\beta, \gamma)$ can be computed as follows:

$$m_{00}(\beta,\gamma) \approx \frac{m_{00}(Q_{11})}{(\beta_2-\beta_1)(\gamma_2-\gamma_1)}(\beta_2-\beta)(\gamma_2-\gamma) + \frac{m_{00}(Q_{21})}{(\beta_2-\beta_1)(\gamma_2-\gamma_1)}(\beta-\beta_1)(\gamma_2-\gamma)$$
$$+ \frac{m_{00}(Q_{12})}{(\beta_2-\beta_1)(\gamma_2-\gamma_1)}(\beta_2-\beta)(\gamma-\gamma_1) + \frac{m_{00}(Q_{22})}{(\beta_2-\beta_1)(\gamma_2-\gamma_1)}(\beta-\beta_1)(\gamma-\gamma_1) \tag{6.22}$$

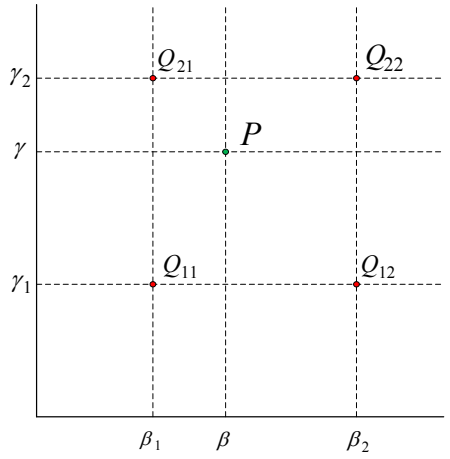


Figure 6-4 Bilinear interpolation

6.1.8 Equation of the planar object in the current camera frame

In Figure 6-3, $F_0 : X_0 - Y_0 - Z_0$ is the camera frame at the desired position and $F_0^{'} : X_0^{'} - Y_0^{'} - Z_0^{'}$ is parallel to F_0. $F_c : X_c - Y_c - Z_c$ is the current camera frame. The equation of the planar object expressed in the current camera frame F_c is given by

$$Z_c = A^{'}(t)X_c + B^{'}(t)Y_c + Z_i^{'}(t) \tag{6.23}$$

Hence depth Z_c of the point on the planar object can be expressed as a continuous function of its image coordinates *x* and y [22], i.e.,

$$\frac{1}{Z_c} = A(t)x + B(t)y + C(t) \tag{6.24}$$

$$A(t) = -\frac{A^{'}(t)}{\lambda Z_i^{'}(t)},\ B(t) = -\frac{B^{'}(t)}{\lambda Z_i^{'}(t)},\ C(t) = \frac{1}{Z_i^{'}(t)} \tag{6.25}$$

where λ is the focal length of the camera. On the other hand, the equation of the planar object if expressed in frame $F_0^{'}$ is given by

$$Z_0^{'} = Z_i \tag{6.26}$$

It is noticed that the rotation matrix from F_c to $F_0^{'}$ is as follows

$$\begin{matrix} F_0^{'} \\ F_c \end{matrix} R_{XYZ}(\gamma,\beta,\alpha) = \begin{bmatrix} c\alpha & -s\alpha & 0 \\ s\alpha & c\alpha & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c\beta & 0 & s\beta \\ 0 & 1 & 0 \\ -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & c\gamma & -s\gamma \\ 0 & s\gamma & c\gamma \end{bmatrix}$$

$$= \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\beta c\beta & s\beta c\beta + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \tag{6.27}$$

where $c\alpha$ is shorthand for $\cos\alpha$, $s\alpha$ for $\sin\alpha$, and so on. Suppose that S is a point on the planar object. If $S(X_0', Y_0', Z_0')$ and $S(X_c, Y_c, Z_c)$ stand for the coordinates of S expressed in F_0' and F_c respectively, one has

$$\begin{bmatrix} X_0' \\ Y_0' \\ Z_0' \end{bmatrix} = \begin{bmatrix} c\alpha c\beta & c\alpha s\beta s\gamma - s\alpha\gamma & c\alpha s\beta c\gamma + s\alpha s\gamma \\ s\beta c\beta & s\beta c\beta + c\alpha c\gamma & s\alpha s\beta c\gamma - c\alpha s\gamma \\ -s\beta & c\beta s\gamma & c\beta c\gamma \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix}$$

From (6.26), one obtains

$$Z_0' = -s\beta X_c + c\beta s\gamma Y_c + c\beta c\gamma Z_c = Z_i$$

i.e., $$Z_c = \frac{s\beta}{c\beta c\gamma} X_c - \frac{s\gamma}{c\gamma} Y_c + \frac{Z_i}{c\beta c\gamma} \tag{6.28}$$

Comparing (6.23) with (6.28), one has

$$A'(t) = \frac{s\beta}{c\beta c\gamma},\ B'(t) = -\frac{s\gamma}{c\gamma},\ Z_i'(t) = \frac{Z_i}{c\beta c\lambda} \tag{6.29}$$

From (6.21) and (6.23), one has

$$A(t) = -\frac{s\beta}{\lambda Z_i},\ B(t) = \frac{c\beta s\gamma}{\lambda Z_i},\ C(t) = -\frac{c\beta c\gamma}{Z_i} \tag{6.30}$$

where *A(t)*, *B(t)* and *C(t)* are used to compute the interaction matrix during visual servoing.

6.1.9 IBVS controller

In 6.1.2, the image features are chosen and rewritten here: $s = \begin{bmatrix} x_g & y_g & a & m_x & m_y & \alpha \end{bmatrix}^T$ and the desired image features are denoted as

$s_d = \begin{bmatrix} x_{gd} & y_{gd} & a_d & m_{xd} & m_{yd} & \alpha_d \end{bmatrix}^T$. Thus the errors of image features are $\Delta s = s - s_d$. A simple proportional control law is given by

$$u = -KJ_{image}^{+}\Delta s \tag{6.31}$$

where u is the camera screw velocity sent to the robot controller, K is the proportional gain, and the J_{image}^{+} is the pseudo inverse matrix of the interaction matrix. The design of visual servoing is accomplished in the following stages

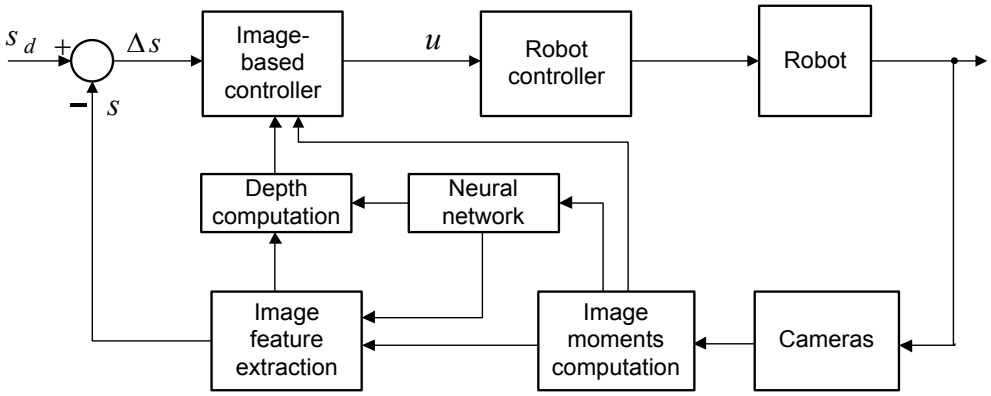


Figure 6-5 Block diagram of the control system

1) Training data acquisition for NN

In this stage, six desired image features s_d of the target object at the desired position are recorded. In the training area, the images of the target object are taken at integral values of β and γ . Then the moment invariants c_1 , c_2 , c_3 , c_4 and the image area $a = m_{00}$ are computed.

2) NN training

In this stage, based on the data sets acquired in the first stage, the NN with inputs of the moment invariants c_1, c_2, c_3, and c_4 are trained.

3) System integration

After being trained, the NN is integrated into the robot control system, and the IBVS robot control signal is computed by (6.31). The block diagram of the robot visual servoing control system is shown in Figure 6-5.

It is noticed that the outputs of NN are the rotational angles around the *x* and *y* axes of the camera frame, which are used to compute two virtual image moments m_x, m_y and to estimate the depth of the 3D target object during visual servoing online. This implies that at any instantaneous moment during visual servoing, the exact depth of the 3D target object can be obtained. Hence, the overall interaction matrix of six image features can be calculated precisely and the performance of visual servoing will be improved.

Experimental Results

The proposed algorithm has been tested on the robot visual servoing tracking system shown in Figure 5-16. The focal length of the camera is λ=3.5 (mm). The desired depth of the object centroid Z_d is 0.15 (m). In the experiments, the desired position and two initial positions of the camera are shown in Figure 6-8. Two objects are tested: one simple central symmetrical planar object and a complex nonsymmetrical planar object. Comparisons with the algorithm of reference [22] and the proposed algorithm with constant depth are given in the experimental results. The first object chosen is a star. The design of NN mapping the relationship between the image moment invariant c_1, c_2, c_3,

c_4 and β , γ is a key step.

Table 6-1 Neural network estimated results

Desired angles (^{o})		Image moment invariants			Estimated angles by NN (^{o})			Errors (^{o})	
β	γ	c_1	c_2	c_3	c_4	$\hat{\beta}$	$\hat{\gamma}$	$\Delta\beta$	$\Delta\gamma$
-15	-15	-1.67	1.86	1.09	95.8	-14.876	-15.109	-0.124	0.109
-15	-10	-1.93	4.31	1.17	127.66	-14.988	-10.006	-0.012	0.006
-15	-5	-2.21	9.64	0.96	408.9	-15.011	-5.01	0.011	0.010
-15	0	-38.07	42.36	26.86	308.41	-15.005	0	0.005	0
-15	5	-2.21	9.64	0.96	408.83	-14.985	4.998	-0.015	0.002
-15	10	-1.93	4.31	1.17	127.66	-14.998	10.003	-0.002	-0.003
-15	15	-1.67	1.86	1.09	95.81	-14.94	15.19	-0.060	-0.19
-10	-15	-1.56	4.32	1.15	13.01	-9.928	-14.944	-0.072	-0.056
-10	-10	-1.84	5.07	1.27	65.21	-9.964	-10.02	-0.036	0.020
-10	-5	-2.04	7.41	1.56	42.42	-9.988	-4.992	-0.012	-0.008
-10	0	-2.19	9.25	4.16	10.76	-9.993	0.005	-0.007	-0.005
-10	5	-2.04	7.41	1.56	42.42	-9.984	5.004	-0.016	-0.004
-10	10	-1.84	5.07	1.27	65.21	-9.936	10.005	-0.064	-0.005
-10	15	-1.56	4.32	1.15	13.01	-10	15.005	0	-0.005
-5	-15	-1.57	1.05	1.02	20.73	-5.022	-15.025	0.022	0.025
-5	-10	-1.78	2.31	1.28	16.55	-5.028	-10.02	0.028	0.020
-5	-5	-1.94	6.83	1.79	18.34	-4.998	-4.9850	-0.002	-0.015
-5	0	-2.03	6.33	5.28	33.54	-5.023	-0.016	0.023	0.016
-5	5	-1.94	6.83	1.79	18.34	-4.99	5.002	-0.010	-0.002
-5	10	-1.78	2.31	1.28	16.55	-4.993	9.999	-0.007	0.001
-5	15	-1.57	1.05	1.02	20.72	-4.999	14.999	-0.001	0.001
0	-15	-1.58	1.14	1.23	52.32	-0.0310	-14.973	0.031	-0.027
0	-10	-1.76	1.42	1.36	43.28	-0.0340	-10.015	0.034	0.015
0	-5	-1.91	3.11	1.54	68.7	-0.001	-4.997	0.001	-0.003
0	0	-1.93	10.51	19.93	70.01	0.0100	0.007	-0.010	-0.007
0	5	-1.91	3.11	1.54	68.7	0	4.997	0	0.003
0	10	-1.76	1.42	1.36	43.28	0.007	9.999	-0.007	0.001
0	15	-1.58	1.14	1.23	52.32	0.0140	15.014	-0.014	-0.014
5	-15	-1.57	1.05	1.02	20.78	5.055	-14.954	-0.055	-0.046
5	-10	-1.78	2.31	1.28	16.55	5.046	-9.947	-0.046	-0.053
5	-5	-1.94	6.83	1.79	18.34	5.005	-4.993	-0.005	-0.007
5	0	-2.03	6.33	5.28	33.54	5.0005	0.001	-0.0005	-0.0007
5	5	-1.94	6.83	1.79	18.34	4.975	5.002	0.025	-0.002
5	10	-1.78	2.31	1.28	16.55	5.001	9.996	-0.001	0.004
5	15	-1.57	1.05	1.02	20..28	5.006	14.987	-0.006	0.013
10	-15	-1.56	4.32	1.15	13.08	10.045	-14.987	-0.045	-0.013
10	-10	-1.84	5.07	1.27	65.12	10.053	-9.988	-0.053	-0.012
10	-5	-2.04	7.41	1.56	42..16	9.984	-5.021	0.016	0.021
10	0	-2.19	9.25	4.16	10.76	10	-0.004	0	0.004
10	5	-2.04	7.41	1.56	42.46	10	4.998	0	0.002
10	10	-1.84	5.07	1.27	65.12	10.012	10.001	-0.012	-0.001
10	15	-1.56	4.32	1.15	1308	10.013	15.001	-0.013	-0.001
15	-15	-1.67	1.86	1.09	95.8	15.052	-14.963	-0.052	-0.037
15	-10	-1.93	4.31	1.17	127.66	15.024	-9.983	-0.024	-0.017
15	-5	-2.21	9.64	0.96	408.82	15.001	-5.023	-0.001	0.023
15	0	-2.41	10.07	0.919	308.06	15.004	0.001	-0.004	-0.001
15	5	-2.21	9.64	0.96	408.71	15.001	5.003	-0.001	-0.003
15	10	-1.93	4.31	1.17	127.67	15.005	10.001	-0.005	-0.001
15	15	-1.67	1.86	1.09	95.8	15.008	15.006	-0.008	-0.006

To train the NN, the configuration of the training data sets acquisition system is shown in

Figure 6-1, which consists of a JAI CM-030 GE camera [87] and a rectangular-shaped object. At the desired position, the object plane is parallel to the image plane of the camera (i.e., β=0, γ=0) and, the depth of the object centroid Z_d =0.15 (m). The range of the rotational angles β and γ around the desired position is ±15º respectively in the training area. The training area is divided into small grids by (1º×1º) in β and γ. The images of the object are taken at the integral values β and γ within the training area. Table 6-1 shows the portion of data sets used for training the NN and the estimation results of the rotational angles around the *x* and *y* axes of the camera frame by the trained NN. Table 6-2 shows the portion of the image area m_{00} corresponding to the integral values β and γ, which is used to determine the depth.

Table 6-2 The image area corresponding to rotational angle

γ / β	-15°	-10°	-5°	0°	5°	10°	15°
-15°	10898	12669	14877	17377	14877	12669	10898
-10°	11542	13296	15439	18017	15439	13296	11542
-5°	11834	13652	15874	18557	15874	13652	11834
0°	12061	13918	16193	18912	16193	13918	12061
5°	11834	13652	15874	18557	15874	13652	11834
10°	11542	13296	15439	18017	15439	13296	11542
15°	10898	12669	14877	17377	14877	12669	10898

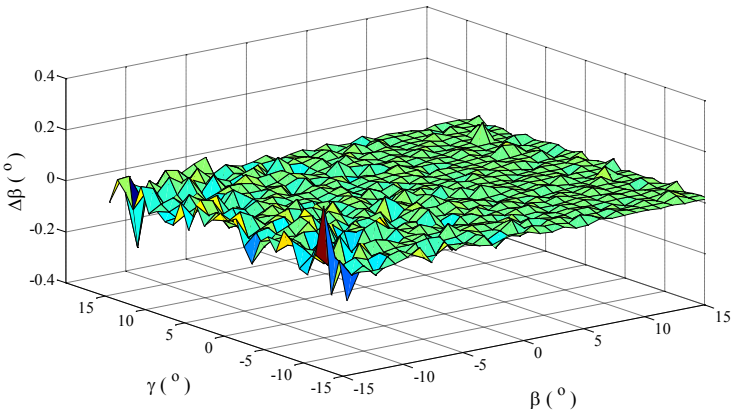


Figure 6-6 Neural network estimation error for β

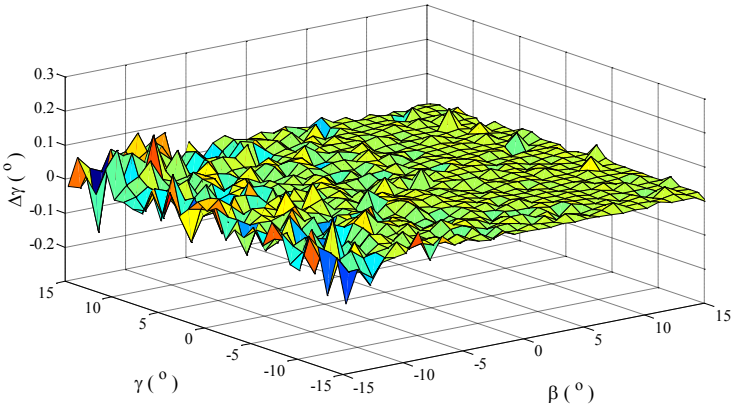


Figure 6-7 Neural network estimation error for γ

In Table 6-1, $\Delta\beta = \beta - \hat{\beta}$, and $\Delta\gamma = \gamma - \hat{\gamma}$ present the estimation errors of β and γ by NN used as the estimator. Figure 6-6 and Figure 6-7 show the estimation errors of β and γ respectively in the training area. From Figure 6-6 and Figure 6-7, one obtains the maximal absolute errors of estimation in the training area, that is $|\Delta\beta_{\max}| = 0.16°$, $|\Delta\gamma_{\max}| = 0.12°$.

For whale-shaped objects, the NN training processes are the same as that of star-shaped objects, which are omitted here. In order to test the proposed scheme, two tasks

with different initial poses for one object are accomplished. The camera's desired position and two initial positions are shown in Figure 6-8. The desired position in the robot base frame is denoted as

$$T_d = \begin{bmatrix} 1 & 0 & 0 & 0.45 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0.15 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

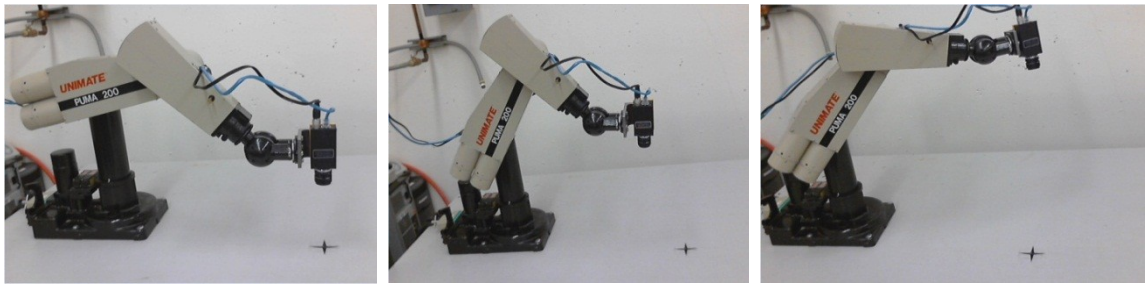
The two initial positions are given as

$$T_{i1} = \begin{bmatrix} 0.9860 & -0.1669 & 0.0038 & 0.5865 \\ -0.1584 & -0.9430 & 0.2926 & -0.3264 \\ 0.0524 & 0.2878 & -0.9562 & 0.8183 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad T_{i2} = \begin{bmatrix} 0.9277 & 0.1706 & 0.3319 & 0.3450 \\ 0.0938 & -0.9674 & 0.2352 & 0.2414 \\ 0.3612 & -0.1871 & -0.9135 & 0.3145 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

As expected, at the desired position, $A(t) = B(t) = 0$, $C(t) = -1/Z_g$, the interaction matrix is rewritten

$$J_{image}^{\parallel} = \begin{bmatrix} -\lambda/Z_g & 0 & x_g/Z_g & x_{g\omega x} & x_{g\omega y} & y_g \\ 0 & -\lambda/Z_g & y_g/Z_g & y_{g\omega x} & y_{g\omega y} & -x_g \\ 0 & 0 & a(2/Z_g) & 3m_{01} & 3m_{10} & 0 \\ 0 & 0 & 0 & c_x & 0 & 0 \\ 0 & 0 & 0 & 0 & c_y & 0 \\ 0 & 0 & 0 & \alpha_{\omega x} & \alpha_{\omega y} & -1 \end{bmatrix}$$

In following experiments, we choose $c_x = c_y = 1$.



(a) Camera desired position (b) Camera initial position 1 (c) Camera initial position 2

Figure 6-8 Desired position and two initial positions

6.1.10 Star-shaped object

Two tasks with different initial poses for the star-shaped object have been accomplished. The initial values and the desired values of image features are provided in Table 6-3. The simulation results are shown in Figure 6-9 to Figure 6-20. The numerical values of the interaction matrix computed for the desired position are given by

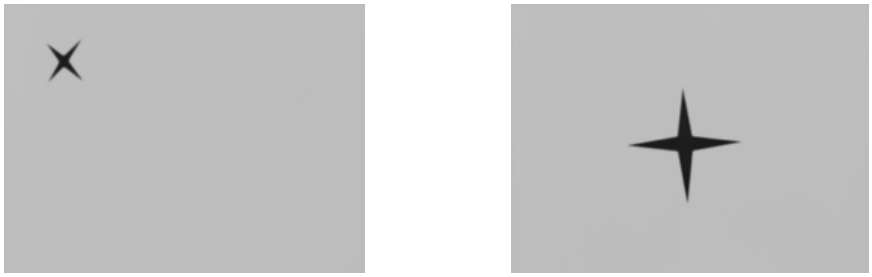
$$J_{image}^{\parallel} = \begin{bmatrix} -0.023 & 0 & 0 & 0 & -1.20 & 0 \\ 0 & -0.023 & 0 & 0.11 & 0 & 0 \\ 0 & 0 & 0.069 & 0.0048 & 0.0083 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.0066 & -0.0011 & -1 \end{bmatrix}$$

It is noticed that its condition number is 1.781, which is very satisfactory.

Table 6-3 The initial values and the desired values of image features for the star-shaped object in task 1 and task 2

Image features	Desired values	Initial values for task 1	Errors of task 1	Initial values for task 2	Errors of task 2
$x_g(pixel)$	0	-212	-212	154	154
$y_g(pixel)$	0	-138	-138	146	146
$a(pixel^2)$	5213	672	-4541	155	-5058
m_x(deg)	0	-11	-11	12	12
m_y(deg)	0	6	6	-7	-7
ϕ(deg)	0	-46	-46	-62	-62
s_x	-0.0456	0.0034	0.049	-0.0686	-0.023
s_y	0.1306	0.1526	0.022	-0.0241	-0.1547

s_x, s_y are the image features in [22].



(a) (b)

Figure 6-9 (a) Desired image and (b) initial image of task 1

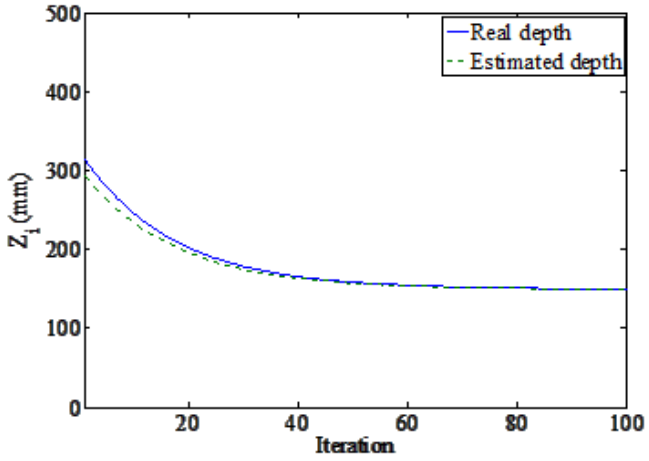
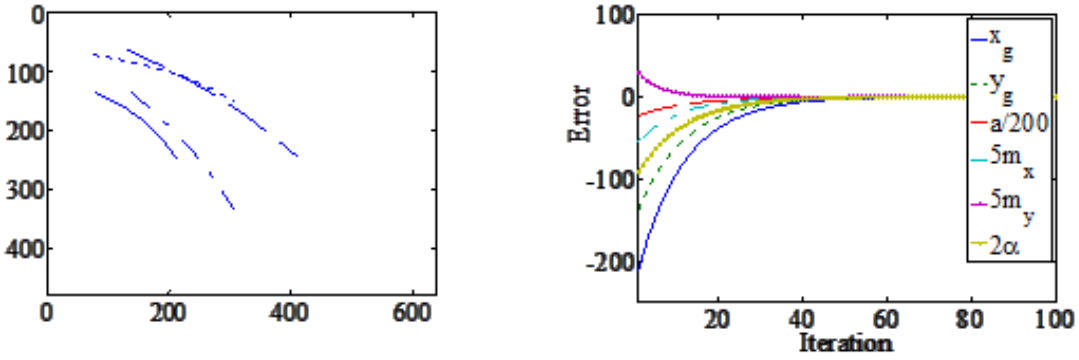
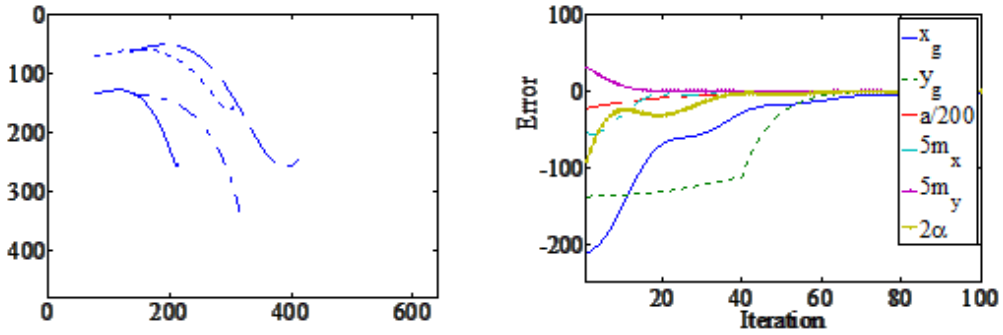


Figure 6-10 Depth estimation result of task 1



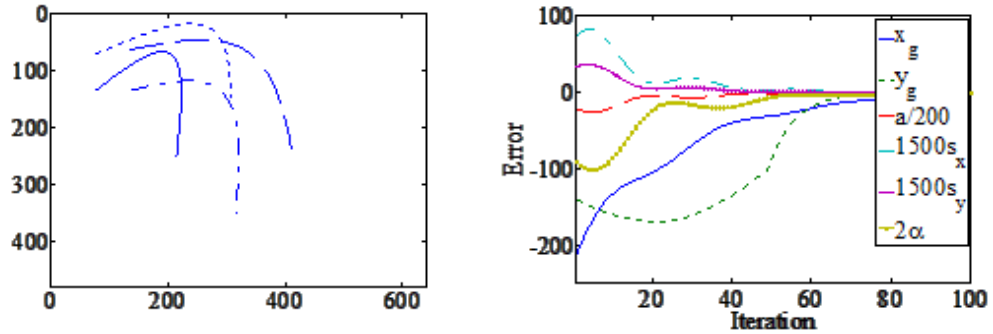
(a) (b)

Figure 6-11 (a) Trajectories of the star's vertices in image frame and (b) Image feature errors of task 1



(a) (b)

Figure 6-12 (a) Trajectories of the star's vertices in the image frame and (b) Image feature errors of task 1 if the depth is constant



(a) (b)

Figure 6-13 (a) Trajectories of the star's vertices in the image frame and (b) Image feature errors of task 1 if controlled by using the algorithm in [22]

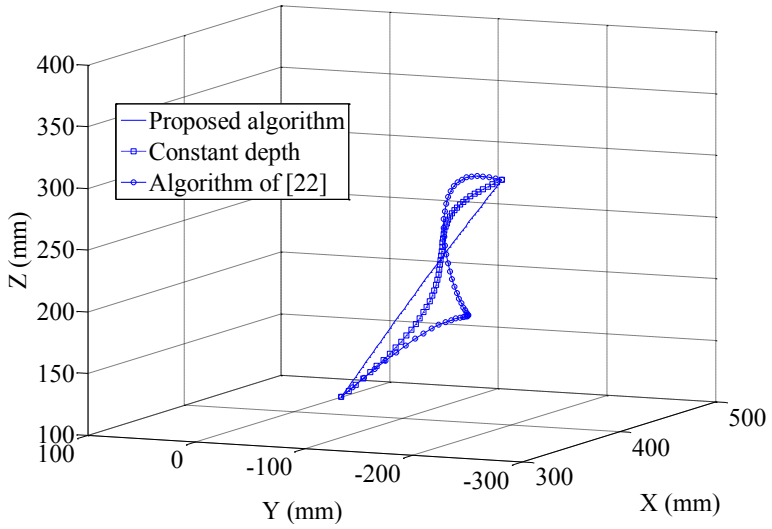
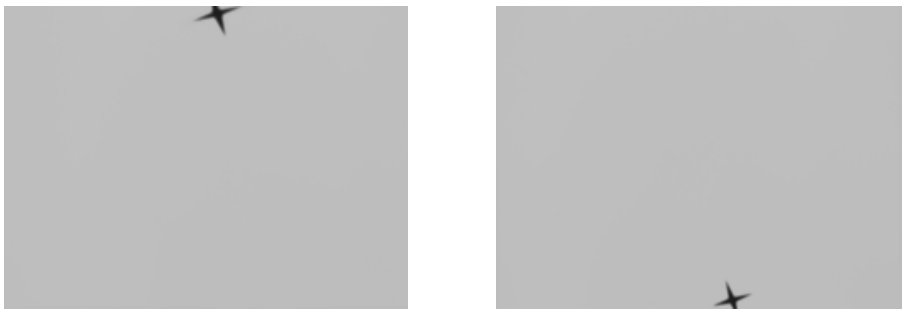


Figure 6-14 Camera trajectories in 3D robot base frame

In task 1, Figure 6-9 (a) and (b) show the desired and initial image positions of the target object respectively. Figure 6-10 shows the depth estimation result. Figure 6-11 (a) and (b) show the trajectories of the vertices of the star-shaped object in the camera image and the image feature errors respectively, controlled by using the proposed algorithm. It is shown that the proposed algorithm obtained a nexponential decrease in image feature errors, and the trajectories of the vertices of the star-shaped object in the camera image frame are nearly a straight line, which results from the exact estimation of the depth and the computation of the interaction matrices of image features. Figure 6-12 (a) and (b) show the experimental results controlled by using the proposed algorithm with a constant depth, which show that the trajectories of the vertices of the star-shaped object in the camera frame are no longer straight lines. Figure 6-13 (a) and (b) show the experimental results controlled by using the algorithm of [22], which show that the trajectories of the vertices of the rectangular-shaped object in the camera frame are complex curves. The camera trajectories in the 3D robot base frame are shown in Figure 6-14. As expected, the camera trajectories of the proposed algorithm in the 3D robot base frame are almost a straight line. The proposed algorithm with a constant depth leads to curved trajectories of the camera in the 3D robot base frame, and the algorithm of [22] leads to a complex curved trajectory of the camera in the 3D robot base frame.

(a) (b)



(c) (d)

Figure 6-15 Initial image (a) and desired image (b) of task 2, (c) and (d) are the images when the object moves out of view of the camera

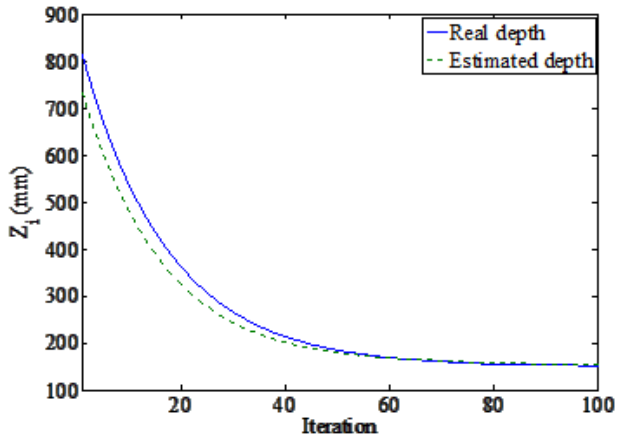
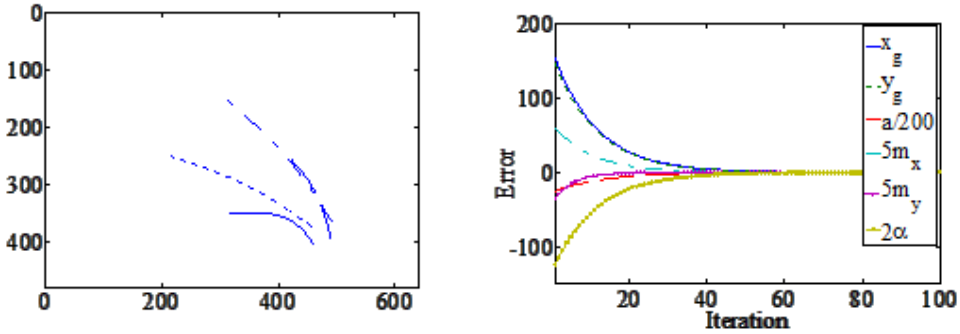
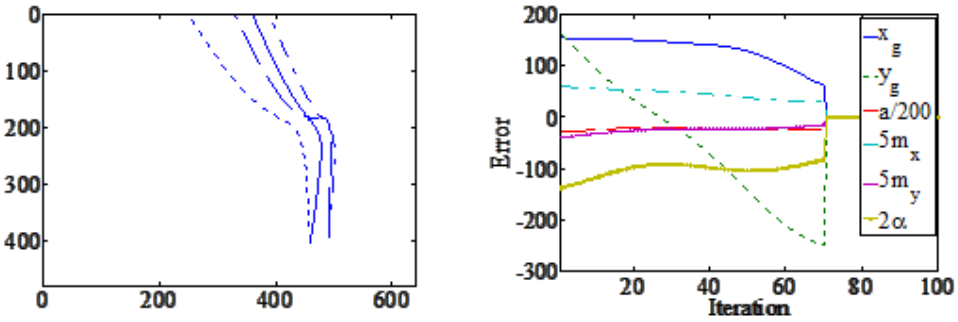


Figure 6-16 Depth estimation result of task 2



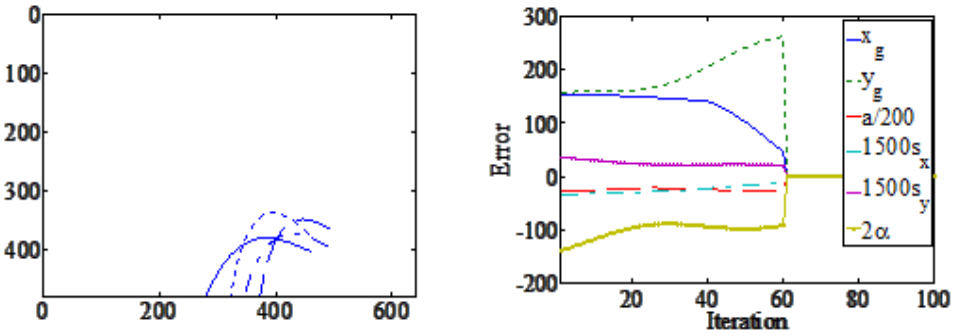
(a) (b)

Figure 6-17 (a) Trajectories of the star's vertices in the image frame, (b) Image feature errors of task 2



(a) (b)

,Figure 6-18 (a) Trajectories of the star's vertices in the image frame, (b) Image feature errors of task 2 if the depth is constant



(a) (b)

Figure 6-19 Trajectories of the star's vertices in the image frame, (b) Image feature errors of task 2 if controlled by using the algorithm in [22]

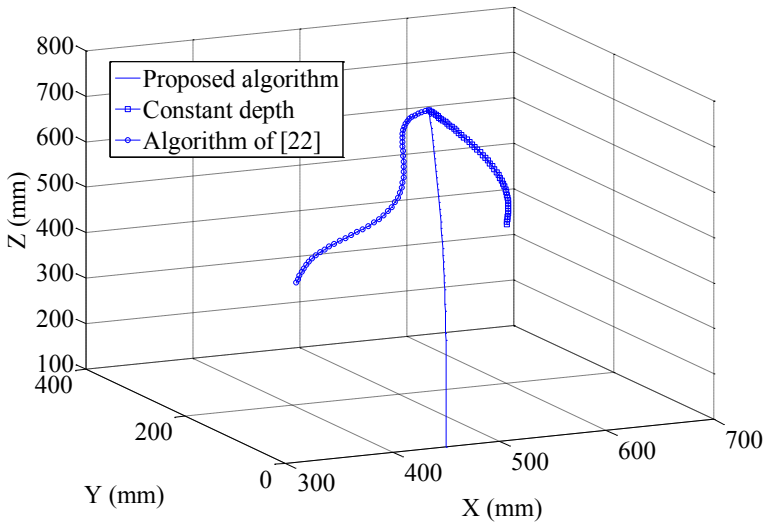


Figure 6-20 Camera trajectories in the 3D robot base frame

In task 2, a large depth is applied in the experiment. Figure 6-15 (a) and (b) show the desired and initial image positions of the target object respectively. Figure 6-16 shows the depth estimation result. Figure 6-17 (a) and (b) show the experimental results of the proposed algorithm. It is shown that the proposed algorithm obtained an exponential decrease in image feature errors, and the trajectories of the vertices of the star-shaped object in the camera image frame are nearly a straight line, due to the exact estimation of the depth and the computation of the interaction matrices. ,Figure **6-18** (a) and (b) show the experimental results achieved by using the proposed algorithm with a constant depth. It is demonstrated that the trajectories of the vertices of the star-shaped object in the camera frame move out of view of the camera and thus lead to the failure of visual servoing. Figure 6-19 (a) and (b) show the experimental results achieved by using the algorithm of [22], which also leads to the failure of visual servoing under the case of large depth. The camera trajectories in the 3D robot base frame are shown in Figure 6-20. As expected, the camera trajectories of the proposed algorithm in the 3D robot base frame are close to a straight line. The proposed algorithm with constant depth, and the algorithm of [22], fail to drive the camera to approach the desired position in the 3D robot base frame.

6.1.11 Whale-shaped object

Two tasks with different initial poses for the whale-shaped object have also been carried out. The initial values and the desired values of image features are provided in

Table 6-3. The experimental results are shown in Figure 6-21 to Figure 6-28. The numerical values of the interaction matrix computed for the desired position are given by

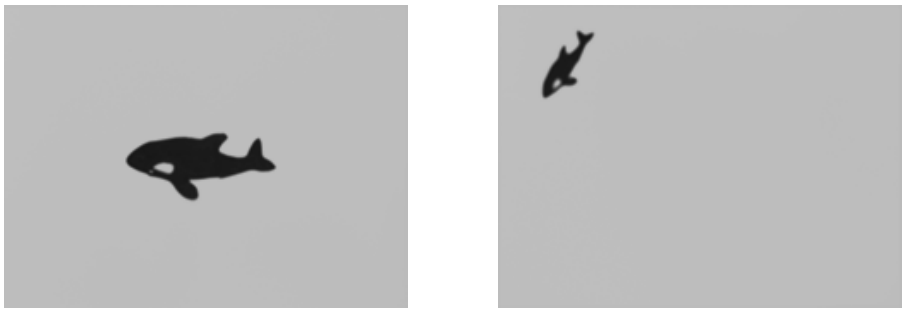
$$J_{image}^{\parallel} = \begin{bmatrix} -0.023 & 0 & 0 & 0 & -0.49 & 0 \\ 0 & -0.023 & 0 & 0.37 & 0 & 0 \\ 0 & 0 & 0.156 & -0.22 & -1.53 & 0 \\ 0. & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.17 & 0.13 & -1 \end{bmatrix}$$

Table 6-4 Initial values and the desired values of image features for a whale-shaped object in task 3 and task 4

Image features	Desired values	Initial values for task 3	Errors of task 3	Initial values for task 4	Errors of task 4
$x_g(pixel)$	0	-215	-215	134	134
$y_g(pixel)$	0	-142	-142	159	159
$a(pixel^2)$	11750	2315	-9435	607	-11143
m_x(deg)	0	-11	-11	12	12
m_y(deg)	0	6	6	-7	-7
ϕ(deg)	0	-63.9	-63.9	39	39
P_x	0.5468	0.6581	0.1113	0.0578	-0.489
P_y	0.0659	0.0734	0.0075	0.0784	0.0125

P_x, P_y are the image features in [22]

It is noticed that its condition number is 2.04, which is satisfactory.



(a) (b)

Figure 6-21 Desired image (a) and initial image (b) of task 3

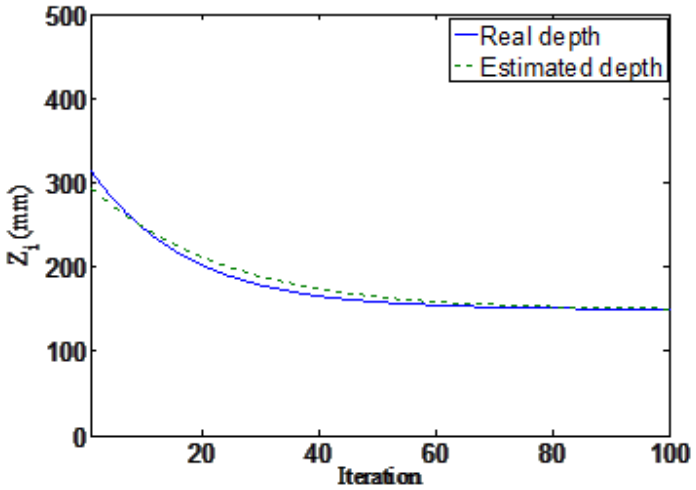
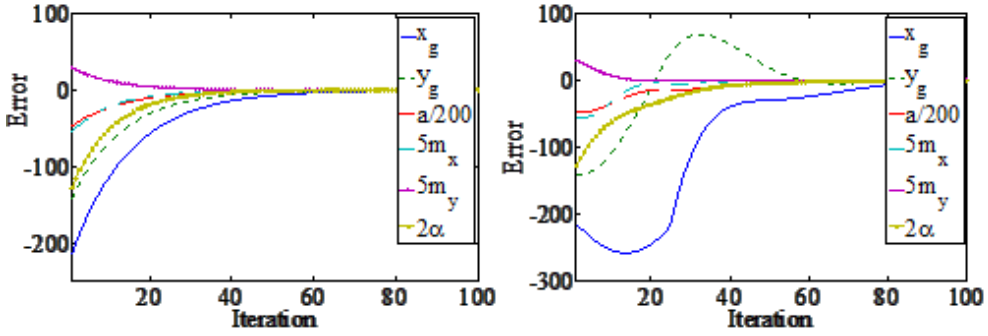
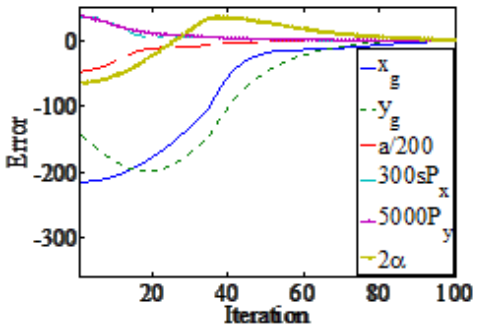


Figure 6-22 Depth estimation result of task 3



(a) (b)



(c)

Figure 6-23 Experimental results of task 3: (a) image feature errors when controlled by the proposed algorithm, (b) the image feature errors if the depth is constant, (c) the image feature errors when controlled by the algorithm in [22]

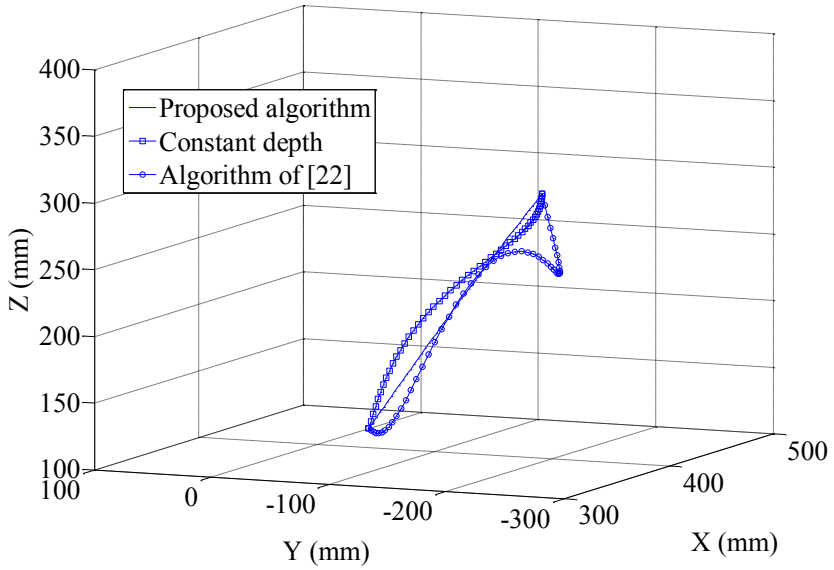
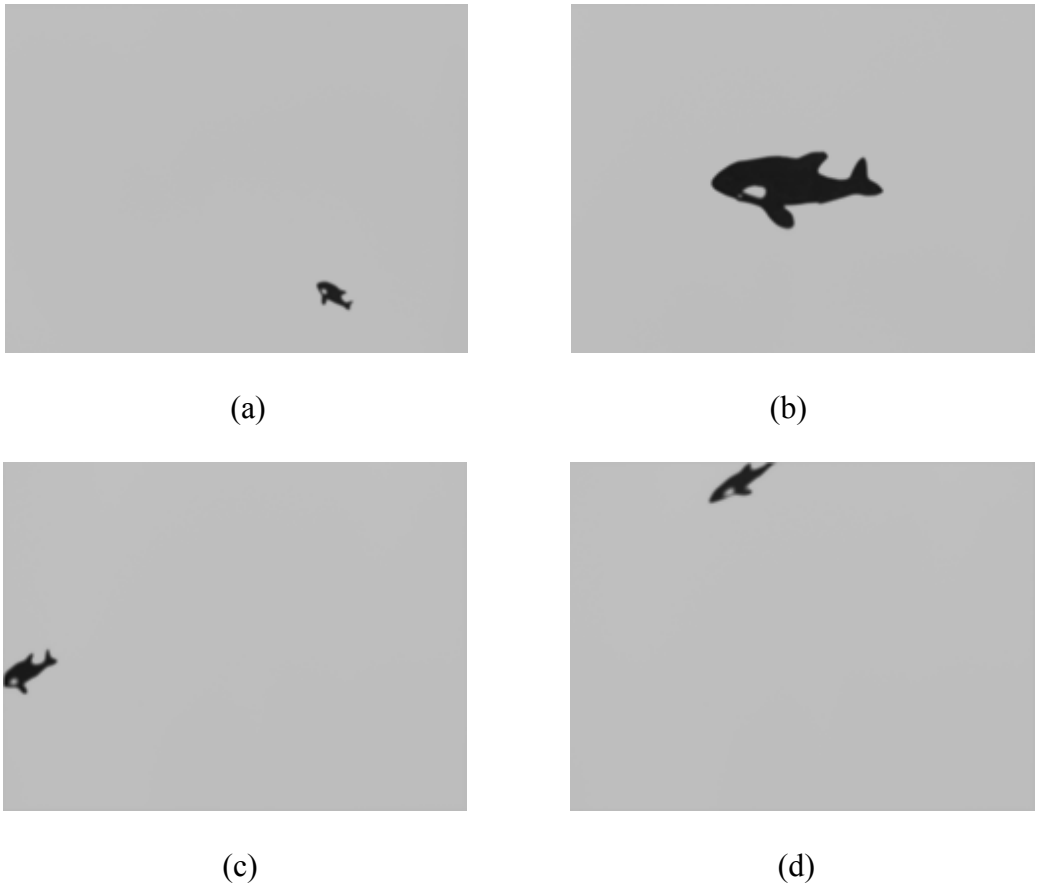


Figure 6-24 Camera trajectories in 3D robot base frame

In task 3, Figure 6-21 (a) and (b) show the desired and initial image positions of the target object respectively. Figure 6-22 shows the depth estimation result. Figure 6-23 (a) shows that the proposed algorithm obtained an exponential decrease in image feature errors due to the exact estimation of the depth and the computation of interaction matrices of the image features. Figure 6-23 (b) and (c) show the experimental results of the proposed algorithm with a constant depth, and of the algorithm of [22]. The camera

trajectories in the 3D robot base frame are shown in Figure 6-24. As expected, the camera trajectories of the proposed algorithm in the 3D robot base frame are like straight lines. The proposed algorithm with a constant depth leads to curved trajectories of the camera in the 3D robot base frame, and the algorithm of [22] leads to complex curved trajectories of the camera in the 3D robot base frame.



(a) (b)

(c) (d)

Figure 6-25 Initial image (a) and desired image (b) of task 4, (c) and (d) the images when the object moves out of view of the camera

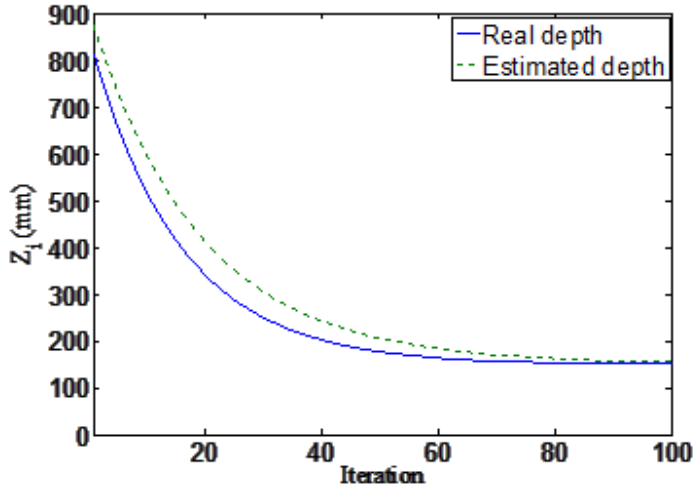
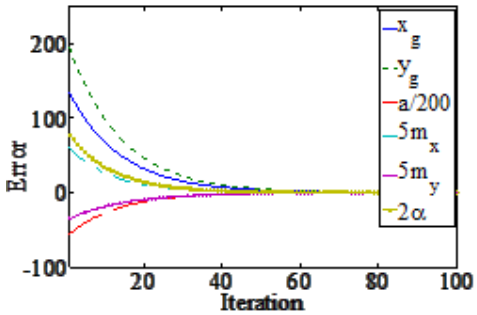
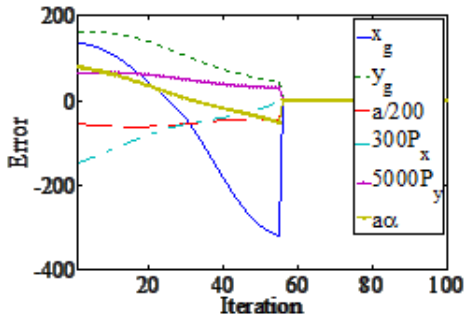


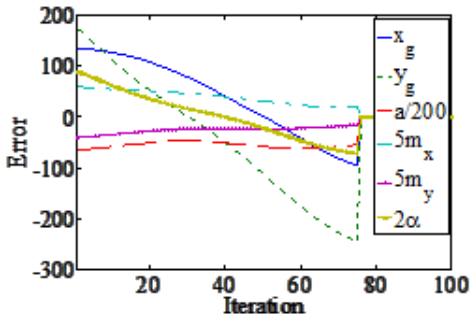
Figure 6-26 Depth estimation result of task 6



(a)



(b)



(c)

Figure 6-27 Experiment results of task 6: (a) image feature errors when controlled by the proposed algorithm, (b) the image feature errors if the depth is constant, (c) the image feature errors when controlled by the algorithm in [22]

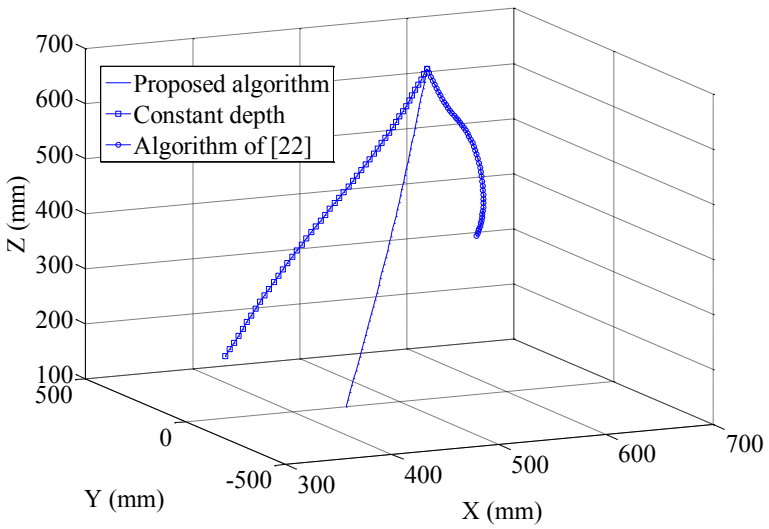


Figure 6-28 Camera trajectories in the 3D robot base frame

In task 4, Figure 6-25 (a) and (b) show the initial and desired positions of the target object respectively. Figure 6-26 shows the depth estimation result. Figure 6-27 (a) shows that the proposed algorithm obtains an exponential decrease in image feature errors due to the exact estimation of the depth and the computation of interaction matrices of the image features. Figure 6-27 (b) and (c) show the experimental results of the proposed algorithm with a constant depth, and of the algorithm of [22]. The camera trajectories in the 3D robot base frame are shown in Figure 6-28. As expected, the camera trajectories of the proposed algorithm in the 3D robot base frame are like straight lines. The proposed algorithm with a constant depth and the algorithm of [22] drives the camera out of the field of view of the camera in the visual servoing process and thus leads to the failure of visual servoing.

The experimental results show that the proposed algorithm can achieve an exponential decrease in image feature errors for different shaped planar objects and from different initial positions. The errors of the image features approach zero smoothly during visual servoing and the image features can be driven to the desired ones in the image frame. In

the 3D world reference frame, the camera can reach the desired position form different initial positions by using the IBVS based on the image moments. The test results on the objects with two different shapes demonstrate that the proposed visual servoing algorithm outperforms the one with a constant depth and the algorithm in [22].

Conclusions

This chapter has presented a new scheme to decouple the angular velocities around the x and y axes of the camera frame in designing the interaction matrix, which can avoid singularities in visual servoing. Based on the neural network, rotation angles around the x and y axes of the camera frame with respect to the desired position of the camera are estimated by the image moment invariants of an object. The theoretical formula to calculate the depth of the target object is derived, which improves the accuracy of the interaction matrix. The interaction matrix related to the proposed image features has a particular decoupled form and the developed IBVS can track the object with different shapes. The experimental results demonstrate the significant improvement in the tracking performance of visual servoing compared with the other algorithms. The robustness of the scheme to the calibration errors and the global stability analysis of the system will be studied in the future.

CHAPTER 7 CONCLUSIONS AND FUTURE WORKS

Conclusions

This dissertation research has addressed challenges in visual servoing for robotic positioning and tracking systems. Two multiple camera configurations of the visual servoing system for robotic positioning are proposed which have potential applications for large-scale robotic manufacturing systems, especially in the aerospace industry. For the robot tracking systems, the improved image moments are proposed as image features for central symmetrical target objects. Based on the chosen image features, the singularity problem of the interaction matrix is solved and the control performance of the visual servoing system is improved. To address the challenge of decoupling the angular velocities around the x and y axes of the camera frame in the interaction matrix, NN-based image moments or virtual image moments are used as image features, which can render a diagonal interaction matrix in visual servoing.

In this research, several visual servoing control strategies have been developed for robotic positioning and tracking systems with different camera configurations. Based upon the research carried out in this dissertation, the following major conclusions are drawn:

1. A multiple camera configuration visual servoing system for large-scale robotic positioning is proposed, which consists of four aligned digital cameras and a 3D CAD laser projector. The 3D CAD laser projector is used to provide precise target points for the visual servoing system at different working positions. The algorithm of

IBVS for floating target points is given and is successfully applied in the simulations of the multiple camera visual servoing system. The simulation results verify the effectiveness of the proposed method and also validate the feasibility of applying multiple camera visual servoing to satisfy the need for large-scale robotic manufacturing systems.

2. A multiple camera-multiple target point visual servoing system is presented. The system includes four aligned digital cameras and eight LEDs that provide four target points and four complementary points. The depths of the target points are estimated online. The dynamic visual servoing controller is designed to drive the end-effector to the desired position from a different initial pose. The simulation results verify the effectiveness of the proposed method, and also validate the feasibility of applying the multiple camera-multiple target point configuration to large-scale robotic manufacturing systems.

3. Two improved image moments related to the target object pose rotating around the x and y axes of the camera frame respectively, are proposed for central symmetrical objects. Simulation shows that the proposed image features can represent the right pose of the target object in visual servoing. Based on the chosen image features, an analytical image interaction matrix is derived and the robot dynamic visual servoing controller is designed based on the derived interaction matrix. The experimental results demonstrate that the tracking performance of the robotic tracking system using the proposed controller has been improved compared to previous methods.

4. Based on NN, a new scheme to decouple the angular velocities around the x and y axes of the camera frame is presented in designing the interaction matrix. The

theoretical formula to compute the depth of the target object is derived, which thus improves the accuracy of the interaction matrix online. Furthermore, the interaction matrix related to the proposed image features has a particular decoupled form, which avoids the singularities of the interaction matrix and therefore remarkably improves the stability of the visual servoing system. The experimental results demonstrate that the tracking performance of the robotic system using the proposed visual servoing control algorithms has been significantly improved.

Future works

The following future works can be conducted to extend the research results presented in this dissertation.

1. The future work on multiple camera visual servoing for large-scale robotic manufacturing systems will focus on:
    - Implementing the developed visual servoing control system on real aerospace manufacturing systems;
    - Stability and robustness studies of multiple camera visual servoing.
2. The future work on the robotic tracking system using the image moments as image features will focus on:
    - Extending the single system visual servoing to the binocular cameras visual servoing system, so as to consider the thickness of an object, and to track the moving object instead of the motionless one;
    - Improving the robustness of the scheme with respect to calibration errors and the global stability analysis of the system.

Publications and patent

The following papers have been published, accepted, or submitted. These papers were written under the guidance of my supervisor, Dr. W. F. Xie. The co-authors have contributed through extensive discussions and setting up the robotic testbed.

Publication list:

1. Y. M. Zhao, W. F. Xie, and X. W. Tu. Performance-based parameter tuning method of model-driven PID control systems. *ISA Trans.*, 51(3):339-399, May 2012.
2. Y. M. Zhao, W. F. Xie, and X. W. Tu. Multiple cameras-multiple target points visual servoing used for robot positioning in large scale 3D manufacturing systems. *Proceedings of The 6th IEEE Conference on Industrial Electronics and Applications (ICIEA 2011),* vol. 2, pp. 2107-2112, Jun 21-23, Beijing 2011.
3. Y. M. Zhao, W. F. Xie, and X. W. Tu. Multiple camera visual servoing for large-scale 3D positioning. In *Proc. of 18th World Congress on Intelligent and Automation*, pp. 462-467, Taiwan, June 21-25, 2011.
4. Y. M. Zhao, W. F. Xie, and X. W. Tu. Improved parameters tuning method of model-driven PID control systems. *Proceedings of The 6th IEEE Conference on Industrial Electronics and Applications (ICIEA 2011)*, vol. 1, pp. 1513-1518, Beijing, Jun 21-23, 2011.
5. Y. M. Zhao, W. F. Xie, and T. T. Wang. Neural network-based image moments for visual servoing of planar objects," *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2012)*, pp. 268-273, Kaohsiung, Taiwan, July 11-14, 2012.

6. T. T. Wang, W. F. Xie, G. D. Liu, and Y. M. Zhao. Quasi-min-max model predictive control for Image-based visual servoing. *IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2012)*, pp. 98-103, Kaohsiung, Taiwan, July 11-14, 2012.
7. Y. M. Zhao, X. W. Tu, and W. F. Xie. Multiple cameras-multiple target points visual servoing in large scale 3D manufacturing systems. Submitted to *Robotics and Computer Integrated Manufacturing (RCIM)*, submitted on March 15, 2011.
8. Y. M. Zhao, and W. F. Xie. Neural network-based image moments for robotic visual servoing. Submitted to *IEEE Trans. on Industrial and Electronics*, Submitted on March 11, 2012.
9. Y. M. Zhao, W. F. Xie, and S. N. Liu. Image-based visual servoing using improved image moments in 6-DOF robot systems. Submitted to *International Journal of Computer Automation System (IJCAS)*. Submitted on May 7, 2012.
10. Y. M. Zhao, W. F. Xie. Multiple cameras visual servoing used for large scale 3D positioning. Submitted to *IEEE Trans. on Robotics and Automation.* Submitted on December 10, 2011.

Patent

W. F. Xie, Y. M. Zhao, and X. W. Tu. Design Visual Servoing System Using Laser Projector for Lager Scale 3-D Pose Correction. Filed for International Patent in NRC (National Research Council, Canada) in June 2010.

REFERENCES

[1] S. Spiess, M. Vincze, M. Ayromiou. On the calibration of a 6-D laser tracking system for dynamic robot measurements. *IEEE Trans. on Instrumentation and Measurement*, 47(4): 270-274, 1998.

[2] J. Hancock, D. Langer. Active laser radar for high-performance measurements. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, pp. 1465-1470, 1998.

[3] S. Hutchinson, G. Hager, and P. I Corke. A tutorial on visual servo control. *IEEE Trans. on Robotics and Automation*, 12(5):651-670, October 1996.

[4] J. Hill and W. T. Park. Real time control of a robot with a mobile camera. In *Proc. of 9th ISIR*, vol. 2, pp. 233-246, Washington, DC, March 1979.

[5] S. Ganpaty. Camera location determination problem. *Technical Memorandum 11358-841102-20-TM*, AT&T Bell laboratories, November 1984.

[6] W. C. Chang. Precise positioning of binocular eye-to-hand robotic manipulators. *Journal of Intelligent Robot System*, 49(1):219-236, 2007.

[7] L. Deng. Comparison of image-based and position-based robot visual servoing methods and improvements. *PhD dissertation of Electrical and Computer Engineering of University of Waterloo*, 2003.

[8] B. Espiau. Effect of camera calibration errors on visual servoing in robotics. In *3rd International Symposium on Experimental Robotics*, Kyoto, Japan, October 1993.

[9] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. In *The Conference of Vision and Control, Series*

Lecture Notes in Control and Information Science, vol. 237, pp. 66-78, Verlag, New York, 1998.

[10] E. Malis, F. Chaumette, and S. Boudet. 2-21/2-D Visual servoing. *IEEE Trans. on Robotics and Automation*, 15(2): 238-250, 1999.

[11] W. F. Xie, Z. Li, X. W. Tu, and C. Perron. Switching control of image based visual servoing with laser pointer in robotic manufacturing systems. *IEEE Trans. on Industrial Electronics*, 56(2): 520-529, 2009.

[12] W. Song, M. Minami. Hand & eye-vergence dual visual servoing to enhance observability and stability. *In Proc. of IEEE International Conference on Robotics and Automation,* pp. 714-721, Kobe, Japan, 2009.

[13] P. I. Corke. Visual control of robot manipulators. *World Scientific Series on Robotics and Automated System*, 2(1): 8-12, 1993.

[14] D. J. Coombs, C. M. Brown. Cooperative gaze holding in binocular vision. *IEEE Control Symposium Magazine*, 11(4): 24-33, 1991.

[15] J. Stavnitzky, D. Capson. Multiple camera model-based 3-D visual servoing. *IEEE Trans. on Robotics and Automation*, 16(6): 732-739, 2000.

[16] W. Song, M. Minami, Y. Mae, and S. Aoyagi. On-line evolutionary head pose measurement by feedforward stereo model matching. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4394-4000, Rome, Italy, 2007.

[17] C. Dune, E. Marchand, and C. Leroux. One click focus with eye-in-hand/eye-to hand cooperation. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2471-2476, Rome, Italy, 2007.

[18] E. J. González-Gávaln, S. R. Cruz-Ramírez, M. J. Seelinger, and J. J. Cervantes-Sánchez. An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision. *Robotics and Computer-Integrated Manufacturing*, 19(5): 387-400, 2003.

[19] E. J. González-Gávaln, S. R. Cruz-Ramírez, M. J. Seelinger, and J. J. Cervantes-Sánchez. An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision. *Robotics and Computer-Integrated Manufacturing*, 19(5): 387-400, 2003.

[20] D. I. Kosmopoulos. Robust Jacobian matrix estimation for image-based visual servoing. *Robotics and Computer-Integrated Manufacturing*, 27(1): 82-87, 2011.

[21] S. Liu, W. F. Xie, and C. Y. Su. Image-based visual servoing using improved image moments. In *Proc. of IEEE International Conference on Information and Automation*, pp. 577-582, Zhuhai, China, June *2009*.

[22] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Trans. on Robotics*, 20(4): 713-723, August 2004.

[23] J. P. Wang and H. Cho. Micropeg and hole alignment using image moments based visual servoing method. *IEEE Trans. on Industrial Electronics*, 5(3): 1286-1294, March 2008.

[24] L. Deng. Comparison of image-based and position-based robot visual servoing methods and improvements. *A PhD dissertation of Electrical and Computer Engineering of University of Waterloo*, 2003.

[25] A. Krupa, J. Gangloff. Autonomous retrieval and positioning of surgical instruments in robotized laparoscopic surgery using visual servoing and laser

pointers. *In IEEE International Conference on Robotics and Automation*, pp. 3769-3774, Washington, DC, May 2002.

[26] C. Collewet and F. Chaumette. A contour approach for image-based control of objects with complex shape," In *Proc. Of IEEE/RSJ International. Conference on Intelligent Robots and Systems.* vol. 1, pp. 751-756, Takamatsu, Japan, November 2000.

[27] P. R. Giordano, A. de Luca, and G. Oriolo. 3D Structure Identification from Image Moments. In of *Proc. of IEEE International Conference on Robotics and Automation*, pp. 93-100, Pasadena, CA, USA, May 2008.

[28] O. Tahri and F. Chaumette. Point-based and region image moments for visual servoing of planar objects. *IEEE Trans. on Robotics*, 21(6): 1116-1127, December 2005.

[29] R. J. Prokop, A. P. Reeves. A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical Models and Image Processing*, 54(5): 438-460, September 1992.

[30] J. Feddema, C. Lee, and O. Mitchel. Automatic selection of image features for visual servoing of a robot manipulator. In *Proc. Of IEEE International Conference on Robotics and Automation*, vol. 2, pp. 832-837, Scottsdale, AZ, May 1989.

[31] B. Nelson and P. Kholsa. The resolvability ellipsoid for visual servoing. In *Proc. of IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 829-832, Seattle, WA, June 1994.

[32] P. I. Corke and S. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Trans. on Robotics and Automation,* 17(1): 507-515, August 2001.

[33] B. Michel, P. Rivers. Singularities in determination of the robot end-effector from the perspective view of points. *Technical Report 1850*, INRIA, 1993.

[34] L. Sicavicco, B. Sicilano. Modeling and control of robot manipulators. *Advanced Textbooks in Control and Signal Processing Series*. 2nd ed., NY: Spring; 2000.

[35] J. Hill and W. T. Park. Real-time control of a robot with mobile-camera. In *9th International Symposium on Industrial Robots*, pp. 233-246, March 1979.

[36] W. C. Chang. Precise positioning of binocular eye-to-hand robotic manipulators. *Journal of Intelligent Robot System*, 49(1): 219–236, 2007.

[37] W. J. Wilson, C. C. W. Hulls, and G. S. Bell. Relative end-effector control using Cartesian position-based visual servoing. *IEEE Trans on Robotics and Automation*, 12(5): 684–696, October 1996.

[38] D. Khadroui, G. Motyl, P. Martinet, J. Gallice, and F. Chaumette. Visual servoing in robotics scheme using a camera/laser-stripe sensor. *IEEE Trans on Robotics and Automation,* 12(5): 743-750, October 1996.

[39] R. C. Velkamp and M. Hagedoorn. State of the art in shape matching. *Technical Report. UU-CS-1999-27*, Utrecht University, Netherlands, 1999.

[40] P. R. Wolf. *Elements of Photogrammetry*. McGraw-Hill, 1974.

[41] J. P. Hespanha. Single camera visual servoing. In *Proc. of 39th IEEE International Conference on Decision and Control.* Sydney, Australia December, 2000.

[42] J. S. C. Yuan. A general photogrammetric method for determining object position and orientation. *IEEE Trans. On Robotics and Automation*, 5(2): 128-142, April 1989.

[43] Y. Yakimovsky and R. Cunninggham. A system for extracting three-dimensional measurement forms a stereo pair of TV cameras. *Computer Graphics and Image Processing*, 7(1): 195-210, 1978.

[44] P. I. Coke. Visual control of robot manipulators. *World Scientific Series in Robotics and automated System*, 2(3): 8-12, 1993.

[45] R. Bukowski, L. S. Haynes, Z. Geng, N. Coleman, A. Santuccim, and K. Lam. Robot hand-eye coordination rapid prototyping environment. In *Proc. Of ISIR*, pp.1615-1628, October 1991.

[46] J. P. Hespanha. Single camera visual servoing. In *Proc. of 39th IEEE International Conference on Decision and Control.* Sydney, Australia December, 2000.

[47] M. H. Korayem and F. S. Heidari. Simulation and experiments for a vision-based control of a 6R robot. *International Journal of Advanced Manufacturing Technology*, 41(3): 367-385, June 2009.

[48] Y. M. Zhao, W. F. Xie, X. W Tu. Multiple cameras-multiple target points visual servoing in large-scale 3D manufacturing systems. In *6th IEEE International Conference on Industrial Electronics and Applications*, pp. 2107-2113, June 2011.

[49] Y. M. Zhao, W. F. Xie, X. W. Tu. Multiple cameras visual servoing used for large scale 3D positioning. In *Proc. of the 8th World Congress on Intelligent Control and Automation***,** pp. 462-467, Taipei, Taiwan, June 2011.

[50] A. Rizzi and D. E. Koditschek. Preliminary experiments in special robot juggling. In *2nd International Symposium on Experimental Robotics*, June 1991.

[51] W. Jang, K. Kim, M. Chung, and Z. Bien. Concepts of augmented image space and for efficient visual servoing of an eye-in-hand robot. *Robotics*, 9(1): 203-212, 1991.

[52] G. S. Srinivasan and J. Dalczynski. Range finder based on intensity gradient measurement. *Applied Optics*, 30(13): 1695-1700, May 9991.

[53] R. C. Harrell, D. C. Slaugher, and P. D. Adsit. A fruit-tracking system for robotic harvesting system. *Machine Vision and Applications*, 2(5): 69-80, 1989.

[54] R. A. Jarvis. A perspective on range finding techniques for computer vision. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(1):122-139, March 1983.

[55] Laser Tracker Manual. *3D metrology and tracking systems*, AMTC, NCR-CNRC, December 2008.

[56] R. S. Ahluwalia and L. M. Fogwell. A modular approach to visual servoing. In *Proc. of IEEE International Conference on Robotics and Automation,* pp. 943-950, 1986.

[57] N. A. Anderrsen, O. Ravn, and A. T. Sorensen. Real-time based control of servo mechanical system. In *2nd International Symposium on Experimental Robotics*, June 1991.

[58] C. H. Andersen, P. J. Burt, and G. S. Van der Wal. Change detection and tracking using pyramid transform techniques. In *Proc. of SPIE*, vol. 579, pp.72-78, Cambridge, Massachusetts, September 1985.

[59] J. S. C. Yuan. F. H. N. Keung and R. A. MacDonald. “Telerobotic tracker”, *EP 0323 681 A1*, European Patent Office, filed 1988.

[60] D. B. Westmore and W. J. Wilson. Direct dynamic control of a robot using an end-point mounted camera and Kalman filter position estimation. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2376-2384, 1991.

[61] R. Nevtia. Depth measurement by motion stereo. *Computer graphics and Image Processing*, 5(1): 203-214, 1976.

[62] A. Rizzi and D. E. Koditschek. Preliminary experiments in special robot juggling. In *2nd International Symposium on Experimental Robotics*, June 1991.

[63] K. A. Dzialo and R. J. Schalkoff. Control implications in tracking moving objects using time-varying perspective-projective imagery. *IEEE Trans. on Industrial Electronics,* IE-33(3)247-253, August 1986.

[64] W. C. Chang, M. L. Chai. Real-time vision-based contour following with laser pointer. In *Proc. of IEEE International Conference on Robotics and Automation*, pp. 2549-2554. September, 2003.

[65] Z. Li, W. F. Xie, and X. W. Tu. Switching control of image based visual servoing with laser pointer in robotic assembly systems. In *IEEE International Conference on Systems, Man and Cybernetics*, pp. 2383-2389, Montreal, Canada, October 2007.

[66] A. Krupa, J. Gangloff. Autonomous retrieval and positioning of surgical instruments in robotized laparoscopic surgery using visual servoing and laser pointers. *In IEEE international Conference on Robotics and Automation*, pp. 3769-3774, Washington, DC, May 2002.

[67] C. Collewet and F. Chaumette. A contour approach for image-based control of objects with complex shape. In *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 1, pp. 751-756, Takamatsu, Japan, November 2000.

[68] B. Yalcin and K. Ohnishi. Infinite-model neural networks for motion control. *IEEE Trans. on Industrial Electronics*, 56(8): 2933-2994, August 2009.

[69] S. M. Gadoue, D. Giaouris, and J. W. Finch. Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers. *IEEE Trans. on Industrial Electronics*, 56(8): 2933-2994, August 2009.

[70] C. L. Xia, C. Gao, and T. N. Shi. A Neural-network-identifier and Fuzzy-controller-based algorithm foe dynamic decoupling control of Premannet-magnet spherical motor," *IEEE Trans. on Industrial and Electronics*, 57(8): 2868-2877, August 2010.

[71] N. J. Cotton, and B. M. Wilamowski. Compensation of nonlinearities using neural networks implemented on inexpensive microcontrollers. *IEEE Trans. on Industrial and Electronics*, 58(3): 733-740, March 2011.

[72] Y. X. Yang, D. Liu, and H. Liu. Robot self-learning visual servoing algorithm using neural networks. In *Proc. of International Conference on Machine Learning and Cybernetics*, pp. 739-742, Beijing, November 2002.

[73] Z. S. Qing, J. Zhao, and F. S. Wang. Using neural network technique in vision-based robot curve tracking. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3817–3822, October 2006.

[74] F. Rvani, G. K. I. Mann, A. Fisher, and R. G. Gosine. Range estimation using TDL neural networks and application to image-based visual servoing. In *IEEE International Conference on Soft Computing in Industrial Applications*. pp. 195-198, Muroran, Japan, June 2008.

[75] R. Bajcsy. Active perception. In *Proc. of IEEE International Conference on Robotics and Automation*, vol.76, no.8, pp.996-1005, August 1988.

[76] A. Djuric, R. A. Saidi, W. ElMaraghy. Dynamic solution of n-DOF global machinery model. *Robotics and Computer-Integrated Manufacturing*, 28(5): 621-630, 2012.

[77] A. Jain, *Robot and Multibody Dynamics: Analysis and Algorithm*. NY: Springer; 2011.

[78] N. Zemiti, G. Morel, A. Micaelli, B. Cagneau, and D. Bellot. On the force control of kinematically defective manipulators interacting with unknown environment. *IEEE Trans. on Control System Technology*, 18(2): 307-22, 2010.

[79] J. J. Craig. *Introduction to Robotics, Mechanics and Control: Nonlinear Control of Manipulators*. Upper Saddle River, 3ed., NJ: Prentice-Hall; 2005.

[80] P. I. Corke. A Robotics Toolbox for MATLAB. *IEEE Robotics and Automation Magazine*, 1(2): 24-32, 1996.

[81] D. H. Ballard, C. M. Brown. *Computer Vision: Stereo Vision and Triangulation*. Englewood Cliffs, NJ: Prentice-Hall; 1982.

[82] M. K. Hu. Visual Pattern Recognition by Moment Invariants. *IRE Trans. on Information Theory*, IT-8(1):179-187, 1962.

[83] B. Yalcin and K. Ohnishi. Infinite-model neural networks for motion control. *IEEE Trans. on Industrial Electronics*, 56(8): 2933-2994, August 2009.

[84] S. M. Gadoue, D. Giaouris, and J. W. Finch. Sensorless control of induction motor drives at very low and zero speeds using neural network flux observers. *IEEE Trans. on Industrial Electronics*, 56(8): 3029-3039, August 2009.

[85] C. L. Xia, C. Gao, and T. N. Shi. A Neural-network-identifier and Fuzzy-controller-based algorithm foe dynamic decoupling control of Premannet-magnet spherical motor. *IEEE Trans. on Industrial and Electronics*, 56(8): 2868-2877, August 2010.

[86] *PUMA 200 Reference Manual*, UNIMATE Inc., 1984.

[87] *CM-030 GE/CB-030 GE User's Manual*, Document Version 2.0, August 2011.

[88] H. Yu, T. T. Xie, S. Paszczynski, and B. M. Wilamowski. Advantages of radial basis function networks for dynamic system design. IEEE Trans. on Industrial and Electronics, 58(12): 5438-5450, December 2011.

[89] http://en.wikipedia.org/wiki/Bilinear_interpolation.